Ontwerp en beheer van ontologieën voor elektronische zorgdiensten

Ontology Design and Management for eCare Services

Femke Ongenae

Promotoren: prof. dr. ir. F. De Turck, prof. dr. ir. T. Dhaene
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Computerwetenschappen

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. D. De Zutter
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2012 - 2013

UNIVERSITEIT GENT

ISBN 978-90-8578-621-4
NUR 984, 981
Wettelijk depot: D/2013/10.500/54

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Informatietechnologie

Promotoren: prof. dr. ir. Filip De Turck
prof. dr. ir. Tom Dhaene

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur

Vakgroep Informatietechnologie
Gaston Crommenlaan 8 bus 201, B-9050 Gent, België

Tel.: +32-9-331.49.00
Fax.: +32-9-331.48.99

Dit werk kwam tot stand in het kader van een
specialisatiebeurs van het IWT-Vlaanderen
(Instituut voor de aanmoediging van Innovatie door
Wetenschap en Technologie in Vlaanderen).

Proefschrift tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen:
Computerwetenschappen
Academiejaar 2012-2013

Dankwoord

Ik zie mezelf nog zitten in Auditorium A van de S9 tijdens de eerste les van het vak Computertechnologie, me luidop afvragend wat die CPU toch was waarover Prof. Hoffman het steeds had. Toen had ik nooit durven dromen dat ik hier vandaag zou staan. In die periode van 10 jaar, heb ik heel wat bijgeleerd, niet enkel op vlak van computerwetenschappen, maar ook op persoonlijk vlak. Zo heeft mijn doctoraat me duidelijk de meerwaarde van interdisciplinair en participatief onderzoek doen inzien. Vanaf die eerste dag heb ik steeds mensen gehad bij wie ik terecht kon voor steun, vragen, een levendige discussie, een aangename babbel, een wijze levensles of gewoon een leuk verzetje. Al deze mensen hebben hun stempel achtergelaten op dit werk en dit boek zou dan ook niet hetzelfde geworden zijn zonder hun bijdrages. Zoals de meesten onder jullie weten, is kort en bondig schrijven niet één van mijn talenten. Dit dankwoord is daar geen uitzondering op. Dus voor zij die niet door onderstaande tekst willen worstelen, hieronder de samenvatting:

Bedankt allemaal!

Allereerst, wil ik mijn promotor Filip De Turck bedanken voor het vertrouwen, de vele technische discussies, de interessante onderzoeksideeën en aangename babbels over kindjes. Je wist het perfecte evenwicht te vinden tussen mij voldoende vrijheid te geven, maar ook op tijd in te grijpen wanneer ik even het bos door de bomen niet meer zag of te veel hooi op mijn vork nam. Ik wil je vooral bedanken om mij vanaf het begin te betrekken en verantwoordelijkheid te geven binnen de verschillende eHealth projecten. Ook mijn co-promotor Tom Dhaene stond altijd paraat om een nieuw licht te werpen op mijn onderzoek. Uw eeuwig optimisme werkte steeds aanstekelijk wanneer het einde van het doctoraat soms zeer veraf leek. Bedankt! Jullie uitmuntende begeleiding en feedback leidde aan het begin van dit onderzoek tot een succesvolle IWT-aanvraag. Hierdoor heb ik mijn doctoraat zonder financiële zorgen kunnen uitvoeren. Mijn dank gaat hiervoor dan ook uit naar het agentschap voor Innovatie door Wetenschap en Technologie (IWT). Ook wil ik Piet Demeester uitdrukkelijk bedanken voor de mogelijkheden die hij mij geboden heeft. Ondanks zijn drukke agenda staat zijn deur altijd open voor grote en kleine vragen. Zijn uitmuntend leiden van de IBCN onderzoeksgroep zorgt voor een aangename werksfeer en laat ons toe om ons volledig te focussen op ons onderzoek.

Daarnaast heb ik het geluk gehad dat 3 mensen zich spontaan ook opgeworpen hebben als mentors gedurende mijn doctoraat. Allereerst wil ik Stijn Verstichel bedanken om mij onder zijn vleugels te nemen en mij de kneepjes van het projectwerk te leren gedurende mijn eerste project, InteGRail. Het was een plezier om die eerste buitenlandse werkreizen met je te ondernemen (al dan niet met een geboekt hotel). Je stond altijd klaar om feedback te geven op mijn onderzoek of mijn Engels, te helpen bij technische problemen, een tandje bij te steken op stresserende momenten of gewoon even een pauze in te lassen met een kopje koffie. Bedankt om al die jaren mijn klankbord te zijn om ideeën tegen af te kaatsen. Ik kijk ernaar uit om de komende jaren nog verder eindeloos met je te discussiëren over ontologieën en eindelijk het debat Amerika vs. Groot-Brittannië voor eens en altijd uit te vechten. Ann Ackaert leerde ik kennen gedurende de tweede helft van mijn doctoraat. Ze dompelde me niet alleen onder in de wondere wereld van de eCare, maar wierp zich ook op als een ware mental coach. Je was er altijd wanneer ik even stoom moest aflaten of je ervaringen over het moederschap of de combinatie werk & gezin te delen. Daarnaast wil ik je zeer erg bedanken om me in contact te brengen met interdisciplinair onderzoek en je vele contacten binnen de gezondheidszorg. Je passie voor het voeren van onderzoek waarbij iedereen, zowel patiënt en zorgverlener als industrie, bij gebaat is, werkt motiverend! Finaal, wil ik Piet Verhoeve bedanken. Al van bij mijn masterproef was hij betrokken bij mijn onderzoek. De brainstormmomenten en feedback gedurende mijn masterproef waren de eerste stappen in de richting van dit doctoraatsonderzoek. Ik wil je bedanken voor de vele brainstormmomenten over de toekomst van de gezondheidszorg, de rol die onderzoek zou moeten spelen in de maatschappij en mij te laten kennis maken met de wondere wereld van verpleegoproepsystemen. Je perspectief vanuit de industrie was van onschatbare waarde voor dit onderzoek.

Tot mijn grootte plezier, heb ik de kans gehad om mijn doctoraatsonderzoek te combineren met een aantal projecten met industriepartners. De verschillende perspectieven en feedback van deze partners hebben een niet te onderschatten impact gehad op mijn onderzoek.

Mijn eerste projectervaring was gedurende de laatste maanden van het InteGRail project. Bedankt Stijn Verstichel en Frederik Vermeulen om me zo snel in te werken in dit grote project. Bedankt ook aan alle andere partners voor de aangename en leerrijke projectvergaderingen, het opsnuiven van de kerstsfeer in München en de kennismaking met Grappa en al het lekkers uit de Middellandse zee in Monopoli. Ook wil ik Frederik, Stijn, Leanneke Loeve en Pieter Dings bedanken voor hun niet aflatende feedback op ons onderzoek en hun medewerking aan het "Transportation" artikel die wel eeuwig in "Revisions Requested"-status leek te blijven steken. De aanhouder wint.

Kort daarna mocht ik ook een handje helpen bij de implementatie van het COSARA project. Bedankt Kristof Steurbaut om samen met mij het medische jargon en de bijhorende databankstructuren te doorgronden. Ik herinner mij nog levendig de vele discussies over de interpretatie van het begrip "dag" op de dienst intensieve zorgen en het urenlang debuggen van de COSARA (of was het COZARA?) monsterqueries. Ook de andere collega's van het project wil ik bedanken voor de aangename samenwerking: Bram Gadeyne, Kristof Taveirne, Kristof Lamont en Sofie Van Hoecke. Het COSARA project startte ook mijn zeer leerrijke

samenwerking met de dokters en verpleegkundigen van het UZ Gent. In het bijzonder wil ik Johan Decruyenaere bedanken. U wist steeds tijd vrij te maken in uw superdrukke agenda als hoofd van de dienst intensieve zorgen om feedback te geven op mijn onderzoek, te brainstormen of om uw unieke inzichten over de informatisering van de gezondheidszorg met mij te delen. Na een meeting met u zit mijn hoofd altijd vol nieuwe ideeën en kan ik niet wachten om deze verder uit te werken. Bedankt voor uw niet aflatende steun aan het eHealth onderzoek binnen IBCN en de mogelijkheid te bieden de resultaten van ons onderzoek te toetsen aan de realiteit in uw dienst. Uw perspectief als dokter op ons onderzoek is van onschatbare waarde. Ook bedankt om me steeds in contact te brengen met uw collega's die mijn onderzoek konden vooruit helpen. Kirsten Colpaert en Dominique Benoit bedankt voor het aanbrengen van de vele nuttige use cases en de diepgaande feedback op onze implementaties. Chris(tian) Danneels bedankt voor de IT ondersteuning en gezellige babbels. Dries Myny bedankt voor het delen van je verpleegkundige ervaring en het ondersteunen van de nodige gebruikersonderzoek en simulaties van het oNCS. Ik kijk ernaar uit om de samenwerking met jullie verder te zetten. Daarnaast wil ik ook nog Stijn Van Looy bedanken voor het aanwakkeren van de machine learning passie en met te wijzen op de problematiek van medische tijdsreeksen. David Verstraeten en Benjamin Schrauwen wil ik bedanken voor de ondersteuning bij het gebruik van ESNs.

Het ACCIO project heeft waarschijnlijk de grootste impact gehad op mijn werk. Al van bij de proposal-fase was ik betrokken bij het project. Het was een plezier om het project volledig mee te realiseren, vanaf de eerste brainstorm tot het succesvol closing event. Bedankt An Jacobs om me onder te dompelen in de wereld van gebruikersonderzoek en me te tonen wat echt interdisciplinair samenwerken kan betekenen. Jouw uniek perspectief heeft me al meerdere malen uitgedaagd om anders na te denken over mijn onderzoek. Bedankt om samen met mij op zoek te gaan hoe we ontologie-gebaseerde gezondheidszorgapplicaties op een gebruikersgedreven manier konden ontwikkelen. Bedankt om de vele "Innovation Binder"-meetings niet enkel nuttig en gefocust te houden, maar daarnaast ook altijd tijd te maken voor wat gezelligheid en de nodige aangename pauzes. Ik kijk ernaar uit om de komende jaren verder uitgedaagd te worden door jou. Ook wil ik Pieter Duysburgh, Lizzy Bleumers, Mathijs Verstraete, Mieke Van Gils, Nicky Sulmon en Annelies Veys en bedanken voor de wederzijdse kennisuitwisseling. Het was een plezier om van jullie gebruikersonderzoek te leren, jullie kennis te laten maken met ontologieën en samen de nieuwe methodologie uit te werken. Bedankt om de hoop niet op te geven gedurende de vele Babylonische spraakverwarringen en het toeslaan van de wet van Murphy gedurende de gebruikersevaluaties. Pieter D, bedankt voor je humor en je relativeringsvermogen. Een meeting met jou eindigt altijd in een positieve noot. Saar De Zutter en Piet V, bedankt voor jullie tomeloze inzet en feedback. Saar zeker bedankt voor je blijvende inzet om de PoC tot een goed einde te brengen en alle eindjes mooi aan elkaar te knopen voor je vertrok op zwangerschapsverlof. Stijn V en Saar, ik kan geen 2 aangenamere mensen bedenken om 2 weken mee opgesloten te worden in een fabriek in Poperinge. Stijn De Pestel, bedankt voor je fantastische inzet en de ijver waarmee je de mobiele applicatie geïmplementeerd hebt. Dit liet ons toe om een mooie PoC neer te zetten. Ook wil ik alle gebruikersgroepen in het zonnetje zetten voor hun inzet tijdens het ACCIO project. Aan alle mensen van het Dominiek Savio Instituut, OLV Ziekenhuis Aalst en AZ Maria Middelares een welgemeende dank u. In het bijzonder bedankt aan Conny Pauwels, Bart Sabbe en Thierry Verplancke voor het geloof in onze doelstellingen en het steeds motiveren van jullie collega's om deel te nemen aan onze workshops. Bedankt ook aan Jan Van Hecke, Piet Berlamont en Valerie Dhaeze voor het beschikbaar stellen van de PRoF ruimte(s), voor de technische ondersteuning tijdens het uitwerken van de PoC en ons steeds te voorzien van de broodnodige koffie. Aan de collega's Stijn V. en Ann A., zeer bedankt voor de fantastische samenwerking tijdens ACCIO. Ik ben eindeloos trots op wat we daar verwezenlijkt hebben. Finaal wil ik ook alle andere mensen bedanken die in meer of mindere mate betrokken waren in het ACCIO project. Jullie inzichten hebben bijgedragen tot het uiteindelijke resultaat en lieten hun stempel achter op dit onderzoek. Bedankt Johan Criel, Laurence Claeys, Hans Crampe, Brecht Stubbe, Peter De Boutte, Henk Vansteenkiste, John Gesquiere, Bernard Pauwels, Geert Casteleyn en zovele anderen.

Het werk in ACCIO mondde uit in heel wat vervolgprojecten, namelijk Fall-Risk, O'CareCloudS en Care4Balance. In deze projecten kreeg ik de kans om samen te werken met heel wat IBCN collega's. Femke De Backere bedankt om de fakkel zo goed van mij over te nemen in het O'CareCloudS project zodat ik met een gerust hart op zwangerschapsverlof kon vertrekken. Femke DB, Stijn V en Ann A, bedankt voor de eindeloze discussies over de betekenis van MCD, MCI en MCK, het willen uitpluizen van en het worstelen met OSGi bundles en fantastische brainstormpartners te zijn. Femke DB, ik kijk nu al uit naar de fantastische PoC die je ongetwijfeld zal afleveren. Blijven gaan! Jeroen Hoebeke, Floris Van den Abeele en Jen Rossey bedankt om mij geduldig in te wijden in de wereld van sensornetwerken en CoAP. Jeroen H tevens bedankt voor de gezellige babbels en mij Masha en de beer te leren kennen. Skyler (en Tibe) zijn je dankbaar! Frederik Van Nieuwenborg en Jan Van Ooteghem bedankt om mij kennis te laten maken met economische modellen, maar ook voor de aangename samenwerking en jullie bereidwilligheid om jullie diep in te werken in de wereld van de kennismodellering. Jelle Nelis bedankt voor de technische ondersteuning en de geduldige uitleg van Diamant/Dyamand ook al loop je soms de muren op van mijn chaotische geest. Naast een aantal industriële partners en personen die ik reeds kende uit het ACCIO project, mocht ik in deze projecten ook heel wat nieuwe personen en partners ontmoeten. Bedankt Pieter Crombez, Tommy Carlier, Kevin De Gryse, Stefan Danschotter, Robby Smet, Maya Van den Broeck, Donaat Van Eynde, Sofie Maes, Jasmien Decancq, Karen Willems, Arne jansen, Jan-Henk Annema, Dimitri Van Landuyt, Eric Moons, Jean-Marc Harvengt, Karin Slegers, Kris Luyten, Bart Vanrumste, Jan Van den Bergh, Steven Vandeput, Griet Depoorter, Kim Wuyts, Maarten Decat en zovele anderen voor de reeds aangename samenwerking. Ik kijk uit naar de eindresultaten van deze projecten.

Birgit Morlion wil ik bedanken voor het ondersteunen en mee uit de grond stampen van al deze eHealth projecten. Ook de rest van de iMinds personeelsleden,

op wie ik doorheen de jaren kon rekenen voor het regelen van administratie en het snel beantwoorden van projectvragen, draag ik een warm hart toe.

Ten laatste mocht ik mijn kennismodelleringsexpertise ook aanwenden in het ASTUTE project. Het was een hele ervaring om mee te mogen draaien in dergelijk groot Europees project. Bedankt Anna Hristoskova voor de leuke semantische discussies, de vele ritjes richting Leuven en Brussel (al dan niet met een omwegje) en rekening te houden met de beperkte mobiliteit van mijn zwangere lijf. Bedankt Elena Tsiporkova voor het vertrouwen in ons onderzoek en om een perfect voorbeeld te zijn van hoe een dergelijk project in goede banen kan geleid worden. Tom Tourwé wil ik bedanken om altijd advocaat van de duivel te spelen zodat we al onze keuzes altijd goed moesten kunnen verantwoorden. Het was een ideale voorbereiding op de verdediging van dit doctoraat. Ik wil ook alle andere partners van het ASTUTE project bedanken om ons te laten kennis maken met het domein van de rampenbestrijding, mee te helpen aan de ontwikkeling van de ontologie en ons steeds uit te dagen op vlak van gebruiksgemak en integratiemogelijkheden van onze implementatie. Bedankt Wim Boffé, Leon Rousseau, Tom Mahieu, Xavier Talpe, Jessie Dedecker, Nicolás González-Deleito, Nadia Ferreira, Bart Adams, Dirk Christianen en alle anderen.

Niet alle brainstormmomenten eindigen in een succesvolle projectaanvraag of zijn momenteel nog in volle bloei. Niettemin hebben deze ontmoetingen ook (reeds) hun stempel gedrukt op dit onderzoek. Vermits ik de voorbije jaren heel wat mensen ben tegengekomen, is het onvermijdelijk dat hier een aantal namen zullen ontbreken. Deze ontbreken echter enkel door vergetelheid en ik kan hen verzekeren dat ik hen daarom niet minder erkentelijk ben. Bedankt Erik Mannens, Davy Van Deursen, Peter Lambert, Sam Coppens, Lucy Hederman, Pieter Baeyaert, Toon Timbermont, Werner van Leekwijck, Heidi Buysse, Jan Dehouck, Alfons Meert, Herman Roelandts, Dirk Van Hyfte, Eva Dierckx, Bert Paepen, Luc Thierens, Liselotte Vandekerckhove, Dirk Van Den Wouwer, Leen Verschraegen en alle anderen die mijn pad kruisten.

Gedurende mijn doctoraat heb ik ook mogen proeven van het onderwijs. Het begeleiden van de practica en de projecten van het vak ODS was steeds een aangename uitdaging. Ik wil dan ook heel het team bedanken voor leuke samenwerking: Filip De Turck, Bart Dhoedt, Kristof Steurbaut, Femke DB, Jeroen Famaey, Steven Latré, Stijn V, Bert Vankeirsbilck, Matthias Strobbe, Samuel Dauwe, Dieter Verslype, Hendrik Moens, Niels Bouten, Tim Verbelen en Klaas Roobroeck. Ook het begeleiden van de vakoverschrijdende projecten samen met Femke DB was een plezier. Mijn hart lag echter het meest bij de één op één begeleiding van thesis- en stagestudenten. De wisselwerking en wederzijdse kennisoverdracht heb ik altijd als een grote meerwaarde ervaren. Doorheen de jaren zijn er heel wat de revue gepasseerd. Sommigen onder hen lieten een significante impact na op mijn doctoraatsonderzoek door het uitwerken van alternatieve onderzoekspistes of het aanleveren van nieuwe ideeën. Ik wil jullie hier allen oprecht voor bedanken! Sommigen onder hen kan ik nu met veel plezier tot mijn collega's rekenen. In het bijzonder wil ik Maxim Claeys en Thomas Vanhove bedanken voor hun bijdrages aan dit onderzoek. Gelukkig stond ik niet alleen in voor het begeleiden van deze

thesisstudenten. Daarom ook een welgemeende dank u wel aan mijn sympathieke co-begeleiders: Femke DB, Stijn V, Thomas Dupont, Wannes Kerckhove, Anna H, Jelle N, Kristof S, Kristof T, Bram G, Jan VO en Frederik VN.

Gezien mijn onhandigheid, ben ik zeer blij dat ik doorheen de jaren kunnen rekenen heb op ons fantastisch admin-team: Jonathan Moreel, Bert De Vuyst, Joeri Casteels, Wouter Adem, Pascal Vandeputte, Johan De Knijf, Bert De Knijf en Serge van Ginderachter. Martine Buysse, Davinia Stevens en Ilse Van Royen stonden altijd paraat voor mijn ontelbare vragen, hulp te bieden bij het doorlopen van de administratieve rompslomp of voor een gezellige babbel bij het halen van een kopje koffie. Martine is de echte moederkloek van IBCN. Bij haar kan je altijd terecht als je even je hart moet luchten. Bedankt hiervoor! Davinia, bedankt voor het gezellig keuvelen over onze kroost en zwangerschapskwaaltjes. Bij vragen rond financiële aspecten kon ik rekenen op de deskundige hulp van Bernadette Becue, Karien Hemelsoen, Nathalie Vanhijfte, Joke Staelens en Dalila Lauwers. Tenslotte wil ik Sabrina en Sandra zeer erg bedanken voor het proper houden van onze werkplaats en klaarzetten van onze vergaderruimtes. Gezien de grootte van IBCN, zijn koekjessmullende en koffiemorsende populatie en de vele technische apparatuur die stof aantrekt als een magneet, is dit zeker geen eenvoudige taak. Sabrina bedankt ook voor de vele geanimeerde en enthousiaste gesprekken op de gang. Je vrolijke “Goeiemorgen” doet mijn ochtendhumeur altijd verdwijnen als sneeuw voor de zon.

Ook de aangename werksfeer was van groot belang. Sinds het begin van mijn doctoraat kan je mij vinden op bureau 2.20. Binnen dit bureau ben ik een paar keer verhuisd om uiteindelijk een vaste plek te vinden tussen mijn twee eilandgenootjes Femke DB en Jeroen F. Jeroen, ik herinner me nog perfect dat Filip me vroeg op welke bureau ik graag wou zitten op de Zuiderpoort. Ik zei dat het mij niet uitmaakte zolang ik maar bij jou op de bureau zat. Ik heb me die beslissing nog geen moment beklaagd sinds we dit avontuur samen aangevat hebben. Het was een plezier om samen deze weg af te leggen. Aangezien je een jaar voorliep op mij was je een ideaal voorbeeld van hoe een doctoraat succesvol kon aangepakt worden. Ik kon steeds bij jou terecht voor mijn vele vragen, hulp en zelfs nalezen van de conclusie van dit boek. Ook bedankt voor de vele gezellige momenten buiten de werkvloer over een lekker etentje, een glaasje en/of een gezelschapsspelletje. Femke DB, ik had wat meer ruimte nodig om jou te bedanken. Lees vooral verder! Daarnaast wil ik zeker ook mijn technical support- en debug-team en West-Vlaamse buddies Wannes Kerckhove en Thomas Dupont bedanken. Zelden heb ik zo’n oprecht behulpzame mensen mogen ontmoeten. Steeds stonden jullie voor me klaar als mijn hardware, software of zelfs mijn hersenen het lieten afweten. Ik prijs mezelf gelukkig dat ik steeds een beroep kan doen op jullie uitgebreide kennis als architectuur-, java- en OSGi-goeroes. Daarnaast zorgden jullie voor de nodige ontspanning tijdens de pauzes en na de werkuren. Het is een plezier om de passie voor muziek, en meer specifiek post-rock, met jullie te delen. Thomas bedankt om me de nodige emo-hitjes te leren kennen (Saves the day op kop). Ze zijn echte guilty pleasures geworden. Wannes bedankt om me bloot te stellen aan alle bands die jij dagdagelijks ontdekt. Het beluisteren van deze muziek, zorgde ervoor dat het schrijven aan een strak tempo kon gebeuren zonder te veel afleiding. Ook bedankt voor de vele gesprekken over pot en pint about the answer to the ultimate question of Life, the Universe, and Everything.

Daarnaast zou ik ook al mijn andere (ex-)bureaugenoten willen bedanken voor de gezellige bureausfeer waardoor ik elke dag met plezier kom werken ondanks mijn ochtendhumeur: Bert V, Gregory Van Seghbroeck, Minh Huu Nguyen, Dirk Gorissen, Hendrik M, Samuel D, Lien Deboosere, Bruno Volckaert, Sofie Verbrugge, Tom Verdickt, Dirk De Schrijver, Ali Farhan Azmat, Pol Dockx, Nico Goeminne en Maryam Barshan. Ook de personen van ex-bureau 3.15 draag ik een warm hart toe, al was het maar om hun lach in te houden tot dat ik buitengekoerst was na mijn spectaculaire valpartij over de onzichtbare draad. Ik heb altijd met veel plezier de bureau-activiteiten (co-)georganiseerd. Na de grote IBCN bureauwissel, konden we op deze activiteiten ook rekenen op het aangename gezelschap van een aantal andere collega's: Stijn V, Steven L, Kristof S, Maxim C, Thomas T, Niels B, Philip L en Bram B. Bedankt iedereen om van deze bureauactiviteiten mooie herinneringen te maken vol hilarische momenten en diepzinnige discussies al dan niet overgoten met de juiste hoeveelheid Ethanol. Ik kijk al uit naar onze volgende activiteit, het pretparkbezoek! Ook wil ik de andere vrouwen bedanken voor de eerste gezellige "Women In ICT"-activiteiten: Anna H, Femke DB, Jolien Coenraets, Marlies Van der Wee, Sofie Lambert, Ann Ackaert, Lisa Pattyn en Sofie Verbrugge. Het is geen staatsgeheim dat ik graag een koffietje drink. Aan de koffieautomaat ontstonden vaak aangename gesprekken. Bedankt Sofie Demeyer, Steven L, Kristof S, Jeroen H, Bert V, Thomas D, Wannes K, Jan VO, Stijn V, Femke DB, Jeroen F, Gregory VS, Jelle N, Anna H, Dieter De Witte en Evy Troubleyn voor deze korte, maar inspirerende en ontspannende pauzes.

Gent is natuurlijk veel meer dan de Zuiderpoort alleen. Tijdens mijn studies heb ik vriendschappen gesloten voor het leven. Aan mijn studievrienden uit de richting informatica, bedankt voor de vele steun tijdens het studeren en de examens. Voor al diegenen die mij voorgingen in het bepalen van een doctoraat, bedankt om de lat zo hoog te leggen! Bedankt voor het delen van de "nerdy" hobbies en de vele gaming- & singstaravondjes en LAN-parties. Bedankt, Femke DB, Jeroen F, Jeroen Janssen, Bart Coppens, Bert Verslyppe, Stéphanie Vanhove, Pascal Torfs, Davy Hollevoet en Niels Neirynck. Lieselot, Sofie en de twee Barten bedankt voor de vele leuke en steunende babbels over een heerlijk kopje koffie. Lieselot bedankt om altijd bereid te zijn om samen met mij het verstand op 0 te zetten en eens goed FOUT te gaan op de tonen van de Nineties. Je oprechte vrolijkheid, openheid, eerlijkheid en optimisme verbazen me keer op keer. Don't ever change! Sarah en Niels bedankt voor de vele leuke uitjes, wederzijdse bezoekjes en gezellige babbels over muziek & Tim Burton. Bij jullie voel ik me altijd welkom! Angelique bedankt voor de vele diepzinnige en levendige discussies over politiek, feminisme en het leven in het algemeen. Ook bedankt om me altijd uit te dagen om nieuwe zaken te ontdekken of nieuwe mensen te leren kennen.

Cristina, my BFF, we don't do emotions. Let's not change that. And yes, I switched to English for this. Thanks for being the strong Buffy to my (season 1 through 3) nerdy Willow. Here are two quotes to describe it all:

What can't we face?
If we're together?
What's in this place
That we can't weather?
There's nothing we can't face
Except for bunnies
– Buffy, The Vampire Slayer - Once more, with feeling

"Y'know, I never really thanked you."
"Oh, yeah, please don't. I don't do thanks. I get all red. Have to bail. It's not pretty."
"Well, then forget that thing. Especially the part where I kind of owe you my life."
"Oh, look! Monkey! And he has a little hat. And little pants."
"Yeah, I see!"
– Buffy, The Vampire Slayer - What's my line, part two

Als we spreken over de nodige verzetjes en leuke afleidingen, dan is er één groep die daarin kampioen is. Enorm bedankt aan de festivalbende en andere Heistse en Brugse vrienden voor de tripjes naar de Ardennen, de verkleedpartijen op het gezelligste Karnaval (met een K!) van België, de doldwaze momenten op Groezrock en andere festivals, het ieder jaar opnieuw perfect inzetten van het nieuwe jaar en alle andere hilarische avonden. Bedankt Tamara, Petra, Frederik, David, Joris, Yannick, Kim, Charlotte, Cherilyn, Christoph, Tommy, Reiline, Mario, Evie, Evy, Sven en Bram. In het bijzonder bedankt aan Petra en Tamara om er altijd te zijn voor mij de voorbije jaren en samen met mij de nieuwe uitdagingen aan te gaan van het "volwassen" worden (huisje, boompje, thuisje). En ook, Skyler kijkt uit naar wat meer vriendjes (hint, hint)!

Ik zou ook mijn familie en in het bijzonder mijn ouders uitvoerig willen bedanken. Mama en papa bedankt om het mogelijk te maken dat ik kon gaan studeren en doctoreren. Niet enkel voor de financiële steun, maar ook de morele zoals de vele tripjes naar Gent, het voorzien van de ideale omgeving zodat ik in alle rust kon studeren, de vele goede zorgen voor Skyler gedurende de laatste maanden van het doctoraat en de constante motivatie. Bedankt voor de steun en advies tijdens de eerste maanden van het moederschap. Broertje, we hadden altijd een zeer andere visie op studeren, maar toch bedankt om me altijd te steunen en te zorgen voor de nodige afleiding met een film of een spelletje Final Fantasy. Bedankt ook aan de rest van mijn familie. We zien elkaar niet zo veel, maar als we elkaar zien is het altijd een gezellige babbel over een lekker kopje koffie. Bedankt Evie, Nadine, Diane, Claudine, Roelf, Henny, Marinka, Johan, Mark en Lilian. Ginette, bedankt om te tonen dat men niet hoeft minder te rocken omdat men ouder wordt. Tot later op het eeuwige festival waar AC/DC altijd de headliner is.

Deze voorlaatste paragraaf reserveer ik voor een gezinnetje van drie die doorheen de jaren echt familie van me zijn geworden. Femke DB en Tibe & Karel Maenhout, ik zou een even dik boek kunnen schrijven als dit doctoraat over hoe

dankbaar ik jullie ben. Al bij het begin van mijn studies stonden jullie voor me klaar om mijn achterstand computerkennis bij te benen, ritjes naar de les (ook al was ik weer eens te laat) of voor een leuk avondje uit. Tibe, kleine kapoen, bedankt voor je vrolijke lach, het samen uitsteken van kattenkwaad, de lieve high-fives, de kusjes en aaitjes voor Skyler en om zo'n mooi voorbeeld te zijn voor mijn zoontje. En ook, Bumba op een paardje! Karel, bedankt voor de vele gezellige babbels, het delen van ervaringen als ouder en het altijd klaarstaan voor ons met woord en daad. Bedankt ook voor je geduld wanneer ik weer eens je vrouwtje haar tijd op eis. Femke, sinds onze zwangerschappen is onze band enkel hechter geworden. Ik ben oprecht trots dat ik jou de meter van mijn kindje mag noemen. Bedankt voor de vele moederschaptips, de gezellige koffiemomenten en etentjes, de gedeelde passie voor eHealth en onderzoek, de gezellige (en steeds minder bange) autoritjes en de superaangename en productieve samenwerking. Ik denk met zeer leuke herinneringen terug aan onze buitenlandse tripjes: het overwinnen van de jetlag en emoties in Perth, de vermoeiende maar mooie fietstocht op Rottnest island, het verliefd worden op Fremantle, de steun tijdens de ziekenhuisbezoekjes, het ploeteren door de sneeuw in Milaan, het staren naar ottertjes in Sea World en het verorberen van chocolate cakes, hamburgers en het riante ontbijt in San Diego. Ik kijk uit naar jouw verdediging. Ik weet nu al dat jij dat fantastisch gaat doen. De Femkes, een begrip, een team, voor altijd! Ook bedankt aan de rest van je familie die me steeds warm ontvangen heeft. Bedankt Kristien, Luc en Lennert. Kristien, nu al bedankt voor wat zeker een fantastische receptie wordt.

Last, but not least, wil ik mijn 2 harares bedanken. Mijn steun en toeverlaat Stephen en mijn kleine astro-piraat Skyler. Woorden schieten te kort om uit te drukken hoe graag ik jullie zie en hoe dankbaar ik jullie ben. Na een dag op het werk, staan jullie me thuis altijd op te wachten met een glimlach, een knuffel en een lekkere maaltijd. Jullie deden me steeds beseffen wat de echte prioriteiten in het leven zijn. Skyler, gedurende deze laatste maanden van mijn doctoraat kon je met één lachje alle stress doen verdwijnen. Bedankt voor de leuke speel- en zingmomenten (Klappen in de handjes, o zo blij, want dat afwerken van het doctoraat, is nu voorbij!). Stephen, sinds de dag dat we elkaar voor de eerste keer ontmoet hebben, stond jij altijd voor me klaar met een eindeloos vertrouwen en geloof dat alles goed komt. Bedankt om me keihard te steunen tijdens mijn studies en de vele tripjes heen en weer naar Gent tijdens de examenperiodes. Bedankt om altijd perfect aan te voelen waar ik nood aan heb, je rust en kalmte en je eindeloos relativeringsvermogen. Bedankt om zo'n perfecte papa te zijn voor onze zoon zodat ik deze laatste maanden met een gerust hart aan mijn doctoraat kon werken. Ook bedankt om me op tijd en stond te verplichten om dat doctoraat te laten voor wat het is en even te genieten, ontspannen en de batterijtjes weer op te laden. Bedankt voor alle mooie momenten en herinneringen van de voorbije jaren: de mooie reizen, de romantische weekendjes, de uitbundige concerten en festivals, de liefdevolle momenten met ons zoontje, de lekkere maaltijd elke dag opnieuw, de gezellige tv-avondjes, Elke dag prijs ik mezelf gelukkig dat ik jou mijn man mag noemen. Ik zie jou en Skyler doodgraag.

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

A-BOX	Assertional Box
ACCIO	Ambient-aware provisioning of Continuous Care for Intramural Organizations
AEON	Automatic Evaluation of ONtologies
AMD	Advanced Micro Devices
ANN	Artificial Neural Networks
APRL	Atiya-Parlos Recurrent Language
ARFF	Attribute-Relation File Format
AUC	Area Under the Curve

B

BFO	Basic Formal Ontology
BPDC	BackProagation DeCorrelation

C

CASP	Context-Aware Service Platform
CBMS	Computer-Based Medical Systems
CDSS	Clinical Decision Support System
CI	Confidence Interval
CIHO	Context-embedded Intelligent Hospital Ontology
CMF	Context Managing Framework
CMSANS	Context Management Service with an Awareness and Notification Service
CoBrA	Context Broker Architecture
COPD	Chronic Obstructive Pulmonary Disease
CPU	Central Processing Unit

CRP-S	C-Reactive Protein Serum
CRP-U	C-Reactive Protein Urine
CT	Computed Tomography
CUO	Centre for User Experience Research

D

DDR3 SDRAM	Double Data Rate Type Three Synchronous Dynamic Random Access Memory
DEUS	Design and Easy Use of wireless Services
DL	Description Logics
DOGMA	Developing Ontology-Grounded Methods and Applications
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering

E

e.g.	exempli gratia - for example
EHR	Electronic Health Record
eHealth	electronic Healthcare
EPR	Electronic Patient Record
ERMHAN	Emilia Romagna Mobile Health Assistance Network
ESN	Echo State Network
EU	European Union

F

FE	Feature Extraction
FMA	Foundation Model of Anatomy
FN	False Negative
FOL	First-Order Logic
FP	False Positive
FS	Feature Selection
FWO-Vlaanderen	Fund for Scientific Research Flanders

G

GB	GigaByte

GCC	Generic Conditional Constraint
GDP	Gross Domestic Product
GFO	General Formal Ontology
GHz	Gigahertz
GUI	Graphical User Interface

H

HCOME	Human-Centered Ontology engineering MEthodology
HTTP	HyperText Transfer Protocol

I

i.e.	id est - in other words
IBBT	Interdisciplinary Institute for Broadband Technology
IBCN	Internet Based Communication Networks and Services
ICT	Information and Communication Technology
ICU	Intensive Care Unit
ID	IDentification
IEEE	Institute of Electrical and Electronics Engineers
IMS	Infection Management System
INTEC	Department of Information Technology
IT	Information Technology
IV	IntraVenous
IWT	Institute for the Promotion of Innovation by Science and Technology in Flanders

L

LFU	Least-Frequently Used
LOD	Linked Open Data
LR	Lineair Regression
LRFU	Least Recently/Frequently Used
LRU	Least Recently Used
LSM	Liquid State Machine

M

Max.	Maximum
MB	MegaByte
MEBN	Multi Entity Bayesian Network
MHz	MegaHertz
MICU/SICU	Medical and Surgical ICU
Min.	Minimum
ML	Machine Learning
ms	Milliseconds

N

N3	Notation 3
NB	Naïve Bayes
NIBPm	Non-Invasive Blood Pressure measurement
Nr.	Number

O

O'Care Platform	Ontology-based Care Platform
OECD	Organization for Economic Co-operation and Development
oNCS	ontology-based Nurse Call System
OSGi	Open Services Gateway initiative framework
OWL	Web Ontology Language
OWL 2 EL	OWL 2 Existential quantification Logic
OWL 2 QL	OWL 2 Query Language
OWL 2 RL	OWL 2 Rule Language
OWL-API	OWL Application Programming Interface
OWL-DL	OWL Description Logics
OWL-S	OWL Services Ontology

P

PC	Personal Computer
PDA	Personal Digital Assistant
PDMS	Patient Data Management System
Ph.D.	Doctor of Philosophy
POSTECH	Pohang university Of Science and TECHnology

PRoF	Patient Room of the Future
Pr-OWL	Probabilistic OWL

Q

QoC-aware	Quality of Context-aware

R

RAM	Random Access Memory
RC	Reservoir Computing
RCToolbox	Reservoir Computing Toolbox
RDF	Resource Description Framework
RF tags	Radio-Frequency tags
RFID	Radio-Frequency IDentification
RNN	Recurrent Neural Network
ROC curve	Receiver Operating Characteristic curve
RPM	Revolutions Per Minute
RQ	Research Question

S

SAPm	Systemic Arterial Pressure measured
SAPS	Simplified Acute Physiology Score
SCB	Semantic Communications Bus
sec.	seconds
Serial ATA	Serial Advanced Technology Attachment
SIRS	Systemic Inflammatory Response Syndrome
SMIT	Studies on Media, Information and Telecommunication
SOA	Service-Oriented Architecture
SOCAM	Service-Oriented Context-Aware Middleware
SPARQL	SPARQL Protocol And Resource Description Framework Query Language
Std. Err.	Standard Error
SUMO	Suggested Upper Merged Ontology
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language

T

T-Box	Terminological Box
TN	True negative
TP	True positive
Turtle	Terse RDF Triple Language

U

UMBEL	Upper Mapping and Binding Exchange Layer
UML	Unified Modeling Language
URL	Uniform Resource Locator
US	United States

W

W3C	World Wide Web Consortium
WBC	White Blood Cell Count
WEKA	Waikato Environment for Knowledge Analysis
WHO	World Health Organization
WSN	Wireless Sensor Network
WWW	World Wide Web

X

XML	eXtensible Markup Language
XSD	XML Schema Definition

Samenvatting – Summary in Dutch –

De afgelopen jaren is de complexiteit van de gezondheidszorg toegenomen als gevolg van een aantal maatschappelijke factoren, bijvoorbeeld het toenemend aantal patiënten per verpleegeenheid, de nood aan meer gespecialiseerde zorg en een gebrek aan verpleegkundig personeel waardoor een efficiënter gebruik van middelen vereist is. Een verdere toename van de complexiteit is te wijten aan het grote aantal van nieuwe technologieën die geïntroduceerd wordt, bijvoorbeeld voor het ondersteunen en automatiseren van administratieve taken, het beheren van data en het observeren van de parameters van een patiënt. Vandaag de dag moet het personeel, zelfs voor het uitvoeren van één enkele taak, verschillende toestellen en applicaties raadplegen en hun geleverde informatie handmatig combineren om voordeel te halen uit de kennis die geleverd wordt door deze technologische apparatuur. Zorgverleners verliezen tijd, missen potentiële patiëntinzichten en hebben geen compleet overzicht van de huidige context doordat de verzamelde data niet automatisch geaggregeerd en geïntegreerd wordt.

Om deze problemen aan te pakken wordt vaak gebruik gemaakt van ambient-intelligente, pervasieve en contextgevoelige technieken. Het gemeenschappelijk kenmerk van deze technieken is dat technologie verdwijnt naar de achtergrond. Sensors en applicaties worden gebruikt om parameters op te meten en te verzamelen over de omgeving en huidige context, terwijl actuatoren aanpassingen doorvoeren in de omgeving op basis van deze informatie. Het gebruik van deze technieken impliceert een nood aan een methode voor het integreren en exploiteren van al deze heterogene informatie zodat dit niet langer de verantwoordelijkheid is van de zorgverlener. Bovendien laat het integreren van deze informatie toe dat intelligente applicaties worden ontwikkeld die gebruik maken van deze beschikbare data om de zorgverleners te ondersteunen in hun dagelijkse activiteiten. Ontologieën worden vaak gebruikt door bestaande contextgevoelige systemen om dit doel te bereiken. Een ontologie is een semantisch model dat formeel de concepten en hun relaties en attributen binnen een bepaald domein beschrijft.

De introductie van contextgevoelige diensten in de gezondheidszorg verloopt echter traag en moeizaam. De grootste klacht is dat gebruikers hun werkpatronen aanzienlijk moesten veranderen om de diensten nuttig te kunnen gebruiken. Dit is te wijten aan het feit dat de diensten onvoldoende gepersonaliseerd zijn, een gebrek aan aandacht voor de gebruiksvriendelijkheid van de diensten, bv. geautomatiseerde en gepersonaliseerde waarschuwingen, en dat er onvoldoende voor gezorgd wordt dat de gebruikers het gevoel hebben dat ze de controle hebben over de technologie. Bovendien wordt er een zeer grote hoeveelheid heterogene data verzameld door het geautomatiseerd observeren van patiëntparameters, uit medische databanken en door het gebruiken van de vele software. De meeste bestaande contextgevoelige gezondheidsapplicaties zijn niet in staat om op een schaalbare manier om te gaan met dergelijke enorme hoeveelheid aan data. Ten laatste wordt in de huidige gezondheidszorgapplicaties vaak geen rekening gehouden met tijdsreeksen. Veranderingen in medische tijdsreeksen bevatten echter belangrijke informatie over de toestand van een zorgbehoevende, bijvoorbeeld een significante achteruitgang van de gezondheid van de zorgbehoevende als gevolg van een complicatie of een nieuwe pathologie.

De belangrijkste onderzoeksbijdrage van dit proefschrift is het ontwerp en beheer van een contextgevoelig, semantisch en zelflerend platform en bijhorende methodes en algoritmes die het mogelijk maken om op eenvoudige manier gezondheidszorgapplicaties te ontwikkelen die toelaten om zorgverleners en -behoevenden te ondersteunen in hun dagelijkse activiteiten en taken. Dit hoofddoel vertaalt zich in de volgende zes specifieke onderzoeksbijdragen.

Een vaak vergeten feit is dat de toepasbaarheid van een contextgevoelig platform sterk afhankelijk is van de correctheid en volledigheid van het gebruikte kennismodel. De meeste ontologieën ontwikkeld voor de gezondheidszorg spitsen zich echter toe op biomedisch onderzoek en worden vooral gebruikt om op een éénduidige manier medische terminologie te definiëren. De eerste bijdrage van dit onderzoek is dan ook de ontwikkeling van een ontologie voor de continue zorg die alle contextinformatie en kennis modelleert die gebruikt wordt in de verschillende gezondheidszorgsectoren, namelijk ziekenhuizen, residentiële zorg en thuiszorg. De ontologie werd op een modulaire manier ontwikkeld om hergebruik aan te moedigen en ervoor te zorgen dat het model makkelijk kan uitgebreid worden met nieuwe kennis. Het model bestaat uit een algemene ontologie, dat kennis bevat die gebruikt kan worden in alle gezondheidszorgsectoren, en twee specifieke ontologieën waarvan één gericht is op residentiële zorg en één toegespitst is op de zorg verleend in ziekenhuizen. De algemene ontologie is verder opgesplitst in zeven delen, de kernontologieën genaamd, die zich elk toespitsen op een specifiek subdomein, bijvoorbeeld het modelleren van taken of profielen.

Om een ontologie te creëren is een goed begrip vereist van het domein dat dit model wenst te beschrijven. Het betrekken van domeindeskundigen in elke stap van de ontwikkeling van de ontologie zorgt voor een betere acceptatie en gebruik van de technologie die gecreëerd wordt met behulp van dit kennismodel. Bovendien zorgt het ervoor dat de ontologie is afgestemd op de dagelijkse werkpraktijken van de zorgverleners. Daarom werd als tweede onderzoeksbijdrage een participatieve methodologie ontwikkeld voor het creëren van ontologieën. Deze methodologie betrekt sociologen, ontwikkelaars van ontologieën en domeindeskundigen, bijvoorbeeld verpleegkundigen, verzorgers en dokters, in elke stap van het ontwikkelingsproces. De methodologie bestaat uit observaties en vijf soorten workshops die de volledige levenscyclus van een ontologie omvatten. Ze is afgestemd op het ontwikkelen van ontologieën voor minder IT-gerichte domeinen waar de belanghebbenden mogelijks niet in staat zijn of niet bereid zijn om de ontologie zelf te ontwikkelen of om veel tijd te wijden aan het ondersteunen van de ontwikkeling. Richtlijnen werden vooropgesteld die detailleren hoe het etnografisch onderzoek en de verschillende workshops moeten georganiseerd worden zodat andere onderzoekers en ontwikkelaars de methodologie eenvoudig kunnen overnemen en gebruiken voor hun eigen doeleinden. De ontwikkelde methodiek werd grondig geëvalueerd door ze te gebruiken om de eerder besproken continue zorg ontologie te ontwikkelen.

Als derde bijdrage werd een contextgevoelig en semantisch platform ontwikkeld, genaamd het O'Care Platform, die toelaat om op een schaalbare, modulaire en performante manier gezondheidszorgapplicaties en -diensten uit te rollen. Om gemakkelijk intelligente applicatie te kunnen ontwikkelen, moet het platform in staat zijn om de betekenis te interpreteren van de enorme hoeveelheid van heterogene gezondheidszorgdata die aangeleverd wordt door de verschillende toestellen en de relevante informatie eruit te filteren. Om ervoor te zorgen dat de verschillende gezondheidszorgtoepassingen enkel de data ontvangen waarin ze geïnteresseerd zijn op dat moment maakt het platform gebruik van een semantische communicatiebus (SCB) in combinatie met de ontwikkelde continue zorg kernontologieën. Het O'Care Platform combineert expressieve OWL-DL redeneertechnieken om de context te interpreteren, gedistribueerd beheer van het contextmodel & -informatie, intelligente filtering van contextinformatie en de gedistribueerde uitrol van de SCB en het gebruik van een cache om de schaalbaarheid te verbeteren. Er werd een grondige evaluatie van de prestaties van de SCB in het gezondheidszorgdomein uitgevoerd met behulp van een illustratief scenario met betrekking tot drie gezondheidszorgtoepassingen, namelijk een geavanceerd verpleegoproepsysteem ondersteund door een lokalisatie- en een domoticacomponent.

Wanneer nieuwe technologie wordt geïntroduceerd, verandert het gedrag van de gebruikers doordat ze zich eraan aanpassen. Bovendien hebben de verschillende omgevingen waarin de technologie wordt uitgerold, bijvoorbeeld verschillende verpleeg- of ziekenhuisafdelingen, mogelijks iets andere verwachtingen over de manier waarop de contextinformatie in rekening wordt gebracht. Het is moeilijk om deze veranderingen in gedrag en kleine nuances in werkpraktijken in rekening te brengen tijdens het ontwikkelen van de technologie. Daarom werd als vierde contributie een zelflerend platform ontwikkeld die toelaat dat contextgevoelige gezondheidszorgapplicaties hun gedrag aanpassen tijdens de uitvoering om zo tot werkelijk gepersonaliseerde diensten te komen. Het ontwikkelde platform bestaat uit de volgende stappen. Eerst wordt het gedrag van de gebruikers en de context waarin het gedrag zich vertoond geregistreerd door een contextmodel, onder de vorm van een ontologie met bijhorende regelgebaseerde contextgevoelige algoritmes. Uit deze data wordt dan historische informatie verzameld door algoritmes die ontbrekende of onjuiste kennis identificeren in de contextgevoelige applicatie. Deze historische informatie wordt vervolgend gefilterd, opgeruimd en gestructureerd zodat het kan gebruikt worden als input voor data mining technieken. De resultaten van de data mining worden gerangschikt en gefilterd door elk resultaat te associëren met een kans die uitdrukt hoe betrouwbaar en nauwkeurig ze is. Deze resultaten en de bijhorende kansen worden dan geïntegreerd in het contextmodel en de dynamische algoritmes. De kansen verduidelijken aan de gebruikers dat deze nieuwe kennis nog niet bevestigd is door doorgedreven evaluatie. Uiteindelijk kunnen deze kansen aangepast worden, namelijk verhoogd of verlaagd worden, ten gevolge van de contextinformatie die verzameld wordt over het gebruik van de nieuwe kennis. Een diepgaande evaluatie van de toepasbaarheid, correctheid en prestaties van het ontwikkelde platform werd uitgevoerd aan de hand van een scenario met betrekking tot het ontdekken van de reden van een verpleegoproep van een patiënt.

Om te illustreren hoe intelligente en performante gezondheidszorgapplicaties kunnen ontwikkeld worden voor het optimaliseren van de continue zorg met behulp van het ontwikkelde O'Care Platform, de continue zorg ontologie en het zelflerend platform, werd een prototype van een zelflerend ontologiegebaseerd verpleegoproepsysteem, genaamd ONCS, ontwikkeld als vijfde contributie. Dit prototype laat toe dat patiënten zich vrij rond bewegen met draagbare en draadloze toestelletjes om verpleegoproepen mee te maken. Bovendien beheert dit platform de profielen van de patiënten en zorgverleners op een efficiënte manier in de continue zorg ontologie. Een innovatief verpleegoproepalgoritme werd ontwikkeld die zich dynamisch aanpast aan de huidige context door de profielinformatie in rekening te brengen. Het algoritme bepaalt eerst de kans dat een oproep een bepaalde prioriteit heeft op basis van de risicofactoren van de patiënt en het type van de oproep. Vervolgens wordt een drempelalgoritme gebruikt om de prioriteit van een specifieke oproep te bepalen op basis van deze kansen. Tenslotte wordt de meest geschikte zorgverlener toegewezen aan de oproep op basis van de beschikbare contextinformatie en de prioriteit van de oproep. Een uitgebreide evaluatie van de toepasbaarheid en prestaties van het ontwikkelde oNCS prototype werd uitgevoerd op basis van enerzijds complexe simulaties gebaseerd op gegevens verzameld over drie verpleegafdelingen van het Universitair ziekenhuis van Gent en anderzijds realistische gebruikerstesten in de patiëntenkamer van de toekomst door gebruik te maken van de ontwikkelde mobiele verpleegoproepapplicatie. Tenslotte werd het zelflerend platform gebruikt om de parameters van het oNCS, namelijk de gedefinieerde drempels en kansen, automatisch aan te passen aan de noden en behoeften van de zorgverleners en -behoevenden.

Tenslotte werd als laatste bijdrage een uitbreiding ontwikkeld van het O'Care Platform, de continue zorg ontologie en het zelflerend platform om medische tijdsreeksen te representeren en verwerken. Een gedetailleerde evaluatie werd uitgevoerd om de voordelen te achterhalen van het gebruik van Echo State Networks in plaats van traditionele classificatietechnieken in combinatie met attribuutextractie en -selectie voor het classificeren van tijdsreeksen.

Summary

In recent years, the complexity in nursing organizations has been on the rise due to societal factors, e.g., the increase of the care unit size and specialized care and the lack of nurse staffing which requires a more efficient use of resources. A further increase of complexity is due to the high amount of new technologies that are being adopted, especially to support administrative tasks, data management and patient monitoring. To take advantage of the information, which is collected by all this technological equipment, the staff have to manually combine and consult several devices, even when carrying out one single task. Due to the fact that the available data is not being integrated and aggregated, caregivers lose time, miss out on potential patient insights and lack a general overview of the situation.

To cope with these problems, ambient-intelligent, pervasive and context-aware techniques are often introduced. The common denominator of these techniques is that the technology will blend into the background of the environment and sensors and actuators will be able to sense and adapt our environment. This implies an emerging demand for the integration and exploitation of the heterogeneous information available from all the devices such that the caregivers no longer play the role of the orchestrator between all these technologies. Moreover, this information integration allows building intelligent applications, which exploit all this available data to support the caregivers in their everyday activities. Ontologies are often employed by existing context-aware systems to realize this goal. An ontology is a commonly agreed on semantic model that formally describes the concepts in a certain domain, as well as their relationships and attributes.

However, the adoption of context-aware services in the healthcare domain is lagging behind what could be expected. The main complaint made by the users is that they had to significantly alter workflow patterns to accommodate the system. This is due to inadequate techniques for personalization of the services, a lack of focus on the soft aspects of interaction, e.g., automated and personalized alerts, and the lack of tackling problems, such as the need of the users for control. Moreover, the amount of heterogeneous data, provided by the monitoring equipment, captured in medical databases and generated by the available software, is vast. Most available pervasive healthcare platforms are unable to cope with this huge amount of heterogeneous data in a scalable way. Finally, time series data is often not taken into account in the current healthcare services and applications. Changes in these medical time series contain important information about the condition of a care receiver, e.g., a relevant clinical deterioration due to a complication or a new pathology.

The main research contribution of this dissertation is the design and development of a context-aware, semantic and self-learning framework and accompanying methodologies and algorithms, which allow the easy development of pervasive healthcare applications that support caregivers and care receivers in their daily activities and tasks. This main goal translates into the following six research contributions.

An often overlooked fact is that the strength of any context-aware platform depends heavily on the correctness and completeness of the used knowledge model. However, most of the developed healthcare ontologies focus on biomedical research and are mainly employed to clearly define medical terminology. Therefore as a first contribution, a continuous care ontology, modeling context information and knowledge utilized across the various continuous care settings, i.e., hospitals, residential care and homecare, was developed. The ontology was developed in a modular fashion to promote re-use and allow easy extension of the model with new knowledge. It consists of a high-level ontology, containing knowledge that is applicable across all continuous care domains, and two low-level ontologies, one focusing on residential care settings and one tuned towards hospital settings. The high-level ontology consists of seven core ontologies, each focusing on a particular sub-domain, e.g., modeling tasks or people profiles.

In order to create an ontology for a particular domain, a good understanding of this domain is required. Including the domain expert in every step of the creation of the ontology facilitates the acceptance of the new technology, which is built using this knowledge model. It also ensures that the ontology is tuned towards the daily work practices of the caregivers. As a second contribution, a participatory ontology engineering methodology was designed, which involves social scientists, ontology engineers and domain experts, e.g., nurses, residential caregivers and doctors, in each step of the development process. The methodology consists of observations and five types of workshops, covering the whole ontology life-cycle. It is tuned towards the development of ontologies for less IT-focused domains, where the stakeholders might not be willing or able to construct the ontologies themselves or attribute a large amount of their time. Guidelines are stipulated, which detail how the ethnographic research and different workshop should be organized, allowing other researchers and developers to easily adopt the methodology. A thorough evaluation of the proposed methodology was performed by applying it to develop the previously discussed continuous care ontology.

As third contribution, a context-aware and semantic platform, called the O'Care Platform, was developed, which allows the scalable, modular and performant deployment of healthcare applications and services. To easily build intelligent applications, the platform must be able to interpret the meaning and adequately filter the relevant information out of the huge amount of heterogeneous care data provided by all the devices and sensors. To ensure that the different healthcare applications and services only receive data that they are interested in at that time, a Semantic Communication Bus (SCB) was used in combination with the developed continuous care core ontologies. The O'Care Platform combines expressive OWL-DL context reasoning, distributed management of the context model and information, intelligent filtering of context information and distributed deployment of the SCB and employment of a cache to increase scalability. A thorough performance evaluation of the SCB in the healthcare domain was performed by an illustrative scenario concerning three healthcare applications, namely a sophisticated nurse call system integrated with localization and home automation services.

When new technology is introduced, the behavior of the users changes to adapt to it. Moreover, different environments, in which the application is deployed, e.g., different nursing units or hospital departments, might have slightly different requirements pertaining to how the context information is taken into account. It is difficult to foresee these changes in behavior and small nuances in workflows at development time. Therefore, a self-learning framework was developed as fourth contribution, allowing context-aware healthcare applications to adapt their behavior at run-time and achieve truly personalized healthcare services. The proposed framework consists of the following steps. First, an ontology-based context model with accompanying rule-based context-aware algorithms is used to capture the behavior of the user and the context, in which it is exhibited. Historical information is then gathered by algorithms that identify missing or inaccurate knowledge in the context-aware platform. This historical information is filtered, cleaned and structured so that it can be used as input for data mining techniques. The results of these data mining techniques are then prioritized and filtered by associating probabilities with the obtained results expressing how reliable or accurate they are. These results and the associated probabilities are then integrated into the context model and dynamic algorithms. These probabilities clarify to the stakeholders that this new knowledge has not been confirmed by rigorous evaluation. Finally, these probabilities are adapted, i.e., in- or decreased, according to context and behavior information gathered about the usage of the learned information. A thorough evaluation of the applicability, correctness and performance of the self-learning framework in the healthcare domain was performed by an illustrative scenario concerning the discovery of reasons for patients' call light use.

To thoroughly demonstrate how intelligent and performant healthcare services can be implemented to optimize continuous care by using the developed continuous care ontology, O'Care Platform and the self-learning framework, a prototype of a self-learning, ontology-based Nurse Call System (oNCS) was developed as fifth contribution. This prototype allows that patients walk around freely with portable, wireless nurse call buttons. Additionally, this platform efficiently manages the profiles of the staff members and the patients by encoding this information into the continuous care ontology. A new nurse call algorithm was developed, which dynamically adapts to the situation at hand by taking this profile information into account. It first determines the probability that a call has a particular priority, based on the type of the call and the risk factors of the patient. Next, a threshold algorithm is executed on the calculated probabilities to determine the priority of a particular call. Finally, the available context information and the priority of the call are used to assign the most appropriate caregiver to a call. An elaborate evaluation of the applicability and performance of the developed oNCS prototype was performed. First, intricate simulations were performed, based on data gathered about three departments at Ghent University Hospital. Second, user tests were conducted in the Patient Room of the Future, using the developed mobile nurse call application to give the invited caregivers a first-hand experience of the oNCS. Finally, the self-learning framework was used to automatically adapt the parameters of the oNCS, i.e., the defined probabilities and thresholds, to the needs and requirements of the caregivers and care receivers.

Finally, extensions of the developed O'Care Platform, continuous care ontology and self-learning framework are proposed to represent and reason with medical time series. A detailed evaluation was performed of the advantages of using echo state networks instead of other traditional classifiers, combined with feature extraction and selection, for time series classification.

1 Introduction

"We are stuck with technology when what we really want is just stuff that works."

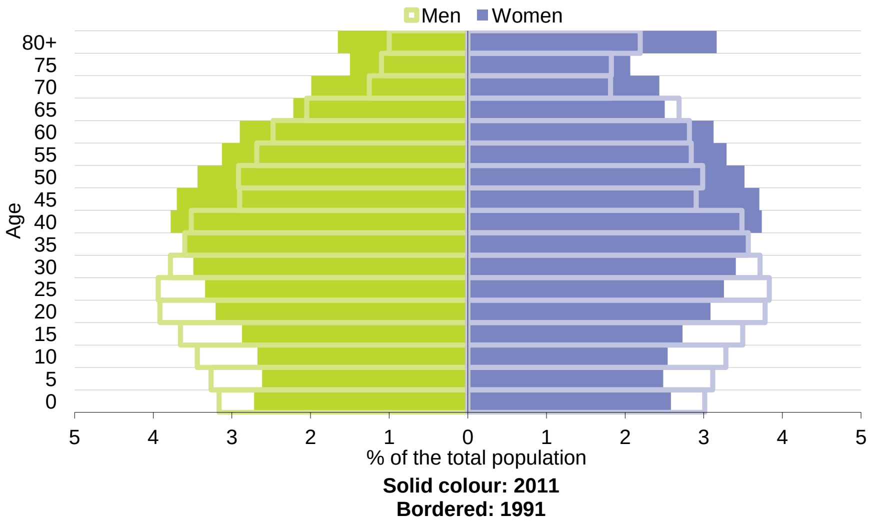
– Douglas Adams, The Salmon of Doubt (1952 - 2001)

1.1 The current healthcare context

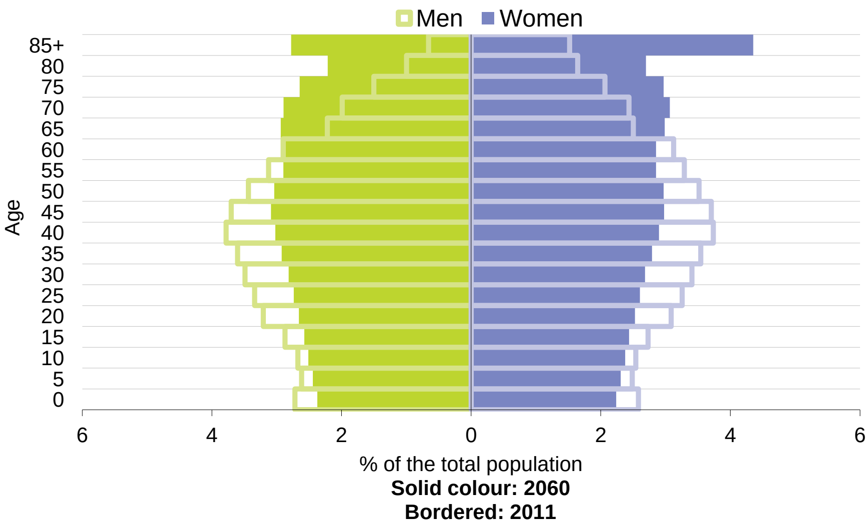
1.1.1 An ageing population and rising costs

Healthcare is the diagnosis, treatment and prevention of disease, illness, injury and other physical and mental impairments in humans. In this dissertation, the people delivering care, e.g., nurses, doctors and informal caregivers, are denoted as caregivers, while the people in need of care are called care receivers, e.g., patients or residents at a nursing home. Healthcare is generally regarded as an important determinant in promoting the general health and well-being of people around the world. According to the World Health Organization (WHO), a well-functioning health care system requires a robust financing mechanism, a well-trained and adequately-paid workforce, reliable information on which to base decisions and policies, and well maintained facilities and logistics to deliver quality medicines and technology [1].

Meeting these requirements becomes increasingly difficult because of two global challenges, namely a) an ageing population and b) the increasing healthcare costs.



(a) The distribution of the EU population by sex and by five-year age groups in 1991 and 2011



(b) Projected distribution of the EU population by sex and by five-year age groups in 2060

Figure 1.1: Age pyramids, source: Eurostat [2]

According to ageing data recorded by Eurostat, as illustrated in Figure 1.1a, the age pyramid of the population belonging to the European Union (EU) is rapidly changing towards a much older population. This is due to consistently low birth rates and a higher life expectancy [2]. It can be noted that, from 1991 until 2011, the share of the population aged less than 15 years in the EU decreased by 3.7%,

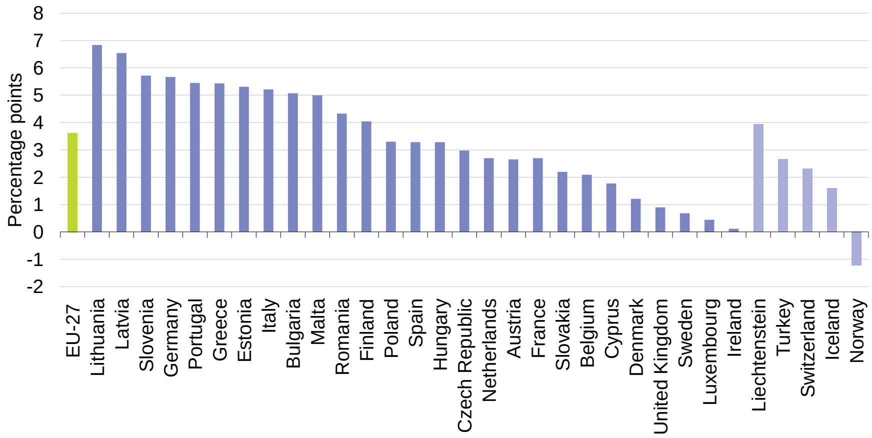


Figure 1.2: The share of the population aged 65 years or over between 1991 and 2011 for the different countries of the EU, source: Eurostat [2]

while the share of the population of 65 years and older increased by 3.6%. As a result, the bottom of the pyramid becomes more narrow, while the top widens. Figure 1.2 shows the share of the population aged 65 years or over between 1991 and 2011 for the different countries of the EU. The share of older people in Belgium increased around 2%. Figure 1.1b shows Eurostat's latest set of population projections, covering the period from 2011 to 2060. It can be derived that the population will continue to age. According to the WHO, the number of people aged 60 years and over is expected to increase from 605 million to 2 billion between 2000 and 2050, while the number of people aged 80 years or older will have almost quadrupled [3]. The number of older people who are no longer able to look after themselves in developing countries is forecast to quadruple by 2050.

The ageing population has a number of socio-economic side effects that influence the provision of healthcare [4]. Because of limited mobility, physical or mental health problems, a lot of elderly are no longer able to live idependently. However, due to the demographic changes and changing family structures, e.g., increased participation of women in the work force, there are less family caregivers, who are able to give informal care to the elderly. Consequently, many require a form of institutionalized long-term care, e.g., residential care, assisted living or long stays in the hospital. These developments are accompanied by emerging staff shortages in the formal care sector. In 2006, the WHO reported an estimated shortage of almost 4.3 million doctors, midwives, nurses and support workers worldwide [5]. Moreover, people are increasingly living longer with chronic diseases. In 2009, more than 133 million Americans, or 45% of the population, have at least one chronic condition [6]. 26% of the Americans have multiple chronic condi-

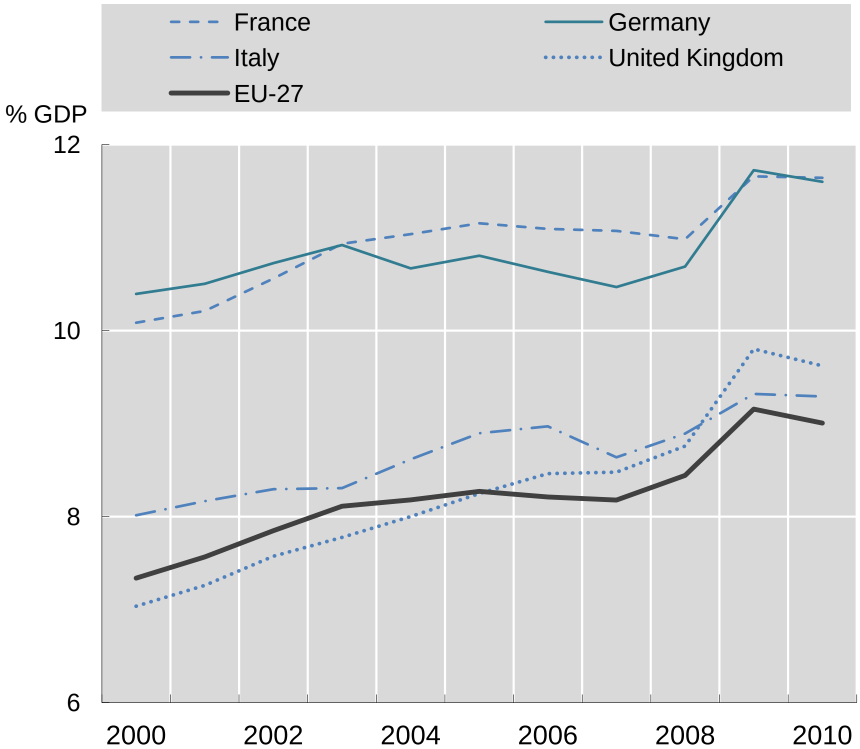


Figure 1.3: Total health expenditure as a share of GDP, 2000-2010, selected EU member states, source: Eurostat, WHO & OECD [7]

tions. This increases the complexity of diagnosis and treatment and requires more personalized healthcare and specialized staff.

Total health expenditure as a share of the Gross Domestic Product (GDP), according to expenditure data recorded by Eurostat, the WHO and the Organisation for Economic Co-operation and Development (OECD) between 2000 and 2010, is shown in Figure 1.3 for selected EU member states [7]. In 2010, EU member states devoted on average 9.0% of their GDP to health spending. This is a significant increase from the 7.3% allocated in 2000, but a slight decline compared to the peak of 9.2% of the GDP in 2009. The latter is due to the economic crisis, which started in many countries in the middle of 2008. Consequently, the healthcare sector is one of the largest industries in most developed countries, bigger generally than education, agriculture, Information Technology (IT), tourism or telecommunications [8]. Spending on healthcare almost invariably grows faster than the GDP [9]. Consequently, we can expect to see the rate of growth of healthcare spending in EU outstrip the GDP growth significantly, during the current economically difficult times. Figure 1.4 visualizes the annual average growth rate in health expenditure

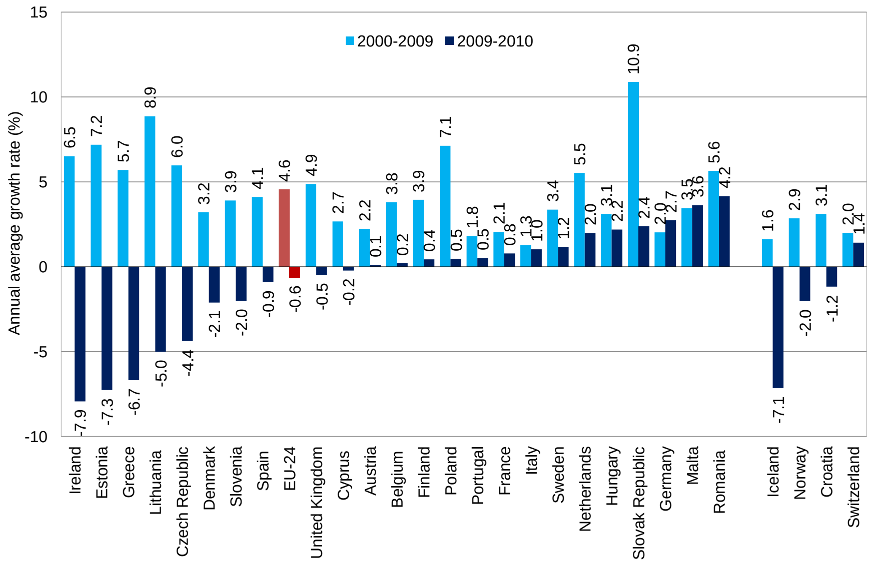


Figure 1.4: Annual average growth rate in health expenditure per capita, in real terms, 2000 to 2010 (or nearest year), source: Eurostat, WHO & OECD [7]

per capita, according to data gathered by Eurostat, the WHO and OECD between 2000 and 2010 [7]. Growth in health spending per capita slowed in 2010 in almost all European countries, reversing a trend of steady increases in many countries. The drop is bigger in countries where the economic crisis hit the hardest. Although, the overall expenditure still increases or stabilizes, the expenditure per capita drops due to the increasing demands on healthcare services by an ageing population.

In summary, there are more elderly people, requiring more complex care, which needs to be provided by a dwindling number of caregivers with an increasingly lower amount of resources per patient.

1.1.2 The rise of eHealth

To achieve a more optimized use of resources and rostering of staff and to reduce the healthcare costs, IT and technological equipment, e.g., monitoring equipment, sensors and nurse call systems, are often introduced in institutionalized healthcare settings [10]. Pagliari et al. [11] performed a very detailed survey of the field of eHealth and came up with the following definition:

> *eHealth is an emerging field of medical informatics, referring to the organisation and delivery of health services and information using the Internet and related technologies. In a broader sense, the term*

characterizes not only a technical development, but also a new way of working, an attitude, and a commitment for networked, global thinking, to improve health care locally, regionally, and worldwide by using information and communication technology.

eHealth is being actively adopted. Major application areas include [12]:

- **Electronic Health Records (EHR)**: A systematic collection of electronic clinical and administrative health information about individual patients or populations for the purpose of supporting administrative tasks, ensuring the continuity of care and performing evidence-based medical research.

- **Telemedicine**: The use of telecommunications to provide medical information and services in situations where the health professional and the patient (or two health professionals) are not in the same location. It includes applications to perform telemonitoring of a patients clinical parameters at home and remote consultation between a health care provider and care receiver.

- **Decision support tools**: Clinical systems that process the wide range of available healthcare data to help health professionals make clinical decisions to enhance patient care.

- **mHealth**: Using mobile devices to collect and aggregate patient data and provide caregivers with a comprehensive overview of this data at the point of care.

- **Healthcare information systems:** Software to ease the administrative healthcare tasks, e.g., appointment scheduling, billing and work schedule management.

This dissertation focuses on the latter three application domains, but also exploits the information captured in EHRs.

The benefits of eHealth, such as improved operational efficiency, higher quality of care, and positive return on investments have been well documented in the literature [13]. Estimations by RAND Corporation predict that costs can be reduced dramatically by the use of information technology. RAND calculated in 2005 that through adoption of health IT by most US hospitals and doctors offices, a potential efficiency saving for both inpatient and outpatient care could average over $ 77 billion per year [14]. The cost saving is attributed to reduced hospital stays, reduced nurses' administrative time and more efficient drug utilization. As such, eHealth provides a way to ensure the continued provision of high quality healthcare to an ageing population at reduced costs and with a more efficient use of resources.

Figure 1.5: Nurse facing ontology orchestration (with a smile!)

eCare is an eHealth research field that focuses on all aspects of care delivery processes. The focus is more on optimizing the continuous care processes, instead of trying to help in the diagnosis of these patients through IT.

1.1.3 The ambient-aware patient room of the future

The increased introduction of technological equipment and eHealth software unfortunately also increases the complexity of healthcare. As visualized in Figure 1.5, caregivers are directly faced with the complex technologies. Today, each technology is being equipped with intelligence on its own, however it is the caregiver that is responsible for orchestrating these technologies and moving information from one technology to another. The caregiver has to use several devices to consult and insert data even when carrying out a single task. This is a very time-consuming job [15]. Due to the fact that the available healthcare data is not being integrated and aggregated, caregivers lose time, lack a general overview of the current situation and miss out on potential insights about care receivers, e.g., early indications of worsening condition of a patient.

The ambient-intelligent patient room of the future contains numerous devices to sense the needs and preferences of all the actors involved and adapts its behaviour accordingly, taking into account the pitfalls of a too static and user depowering system, which gives the users very limited control [16, 17].

Consider, for example, a patient with a concussion, who needs to be in a dark environment. Today, the staff members are responsible for switching on the lights at the appropriate level each time they enter the room. Consequently, each staff member has to be aware of all the aspects and specific characteristics of the patient's condition. If an uninformed person enters the room or a wrong button is pressed, this can cause physical pain for the patient. However, if the lighting control system would be aware of the patient's pathology and needs, it can automatically turn on the light to the correct level when it detects that the nurse enters the room. Moreover, a message can be displayed, explaining to the nurse why the lights are lit on this lower level. Staff members are able to overrule the system, but a light sensor could be used to monitor the light intensity in the room and alert a nurse if a pre-defined threshold is crossed.

To realize this vision, ambient-intelligent, pervasive and context-aware techniques can be adopted. The common denominator of these techniques is that the technology will blend into the background of the environment and sensors and actuators will be able to sense and adapt the environment by interpreting various contexts of an environment and its users [18, 19]. Dey and Abowd [20] refer to context as "any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves". The collected context information is often ambiguous and heterogeneous. This implies an emerging demand for the integration of this avaiable heterogeneous information in a context model, such that intelligent applications can exploit this knowledge. Ontologies are often employed by existing context-aware systems to realize this goal [21–23].

The design and use of ontologies in the healthcare domain to optimize the delivery of healthcare services through context-aware and pervasive systems is an active research field [24, 25], as it has been recognized that ontology-based systems can be used to improve the management of complex health systems [26]. This dissertation builds further on this research, as highlighted in the contributions in Section 1.4. Therefore, a short introduction into ontologies and reasoning is given in the following section.

1.2 Background on ontologies and reasoning

A short, but comprehensive definition of an ontology is given by Gruber [27]: "An ontology is a specification of a conceptualization in the context of knowledge description". An ontology is thus a commonly agreed on semantic model that formally describes the concepts in a certain domain, their relationships and attributes.

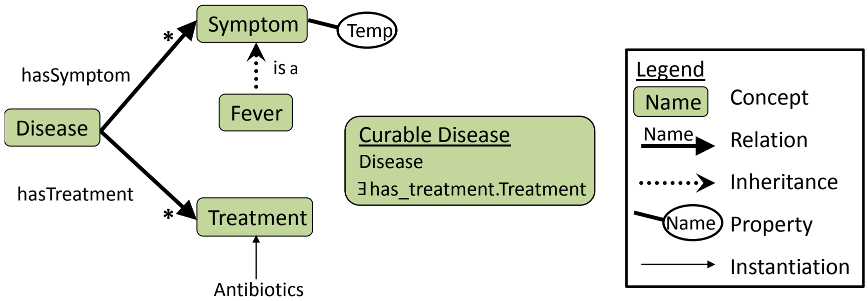


Figure 1.6: Example ontology, modelling diseases and their symptoms

This common, agreed data-format can then be used to exchange the data and its attached domain model, which encourages re-use, communication, collaboration and integration. By managing the data about the current context in an ontology, intelligent algorithms can be more easily defined that take advantage of this information to optimize and personalize the context-aware applications.

An example ontology, that models diseases and their symptoms, is shown in Fig. 1.6. The squares represent the different concepts within the domain, e.g., `Disease` or `Symptom`. The dotted arrows indicate subclass relationships. For example, the subclass relationship between `Fever` and `Symptom` indicates that `Fever` is a kind of `Symptom` and that each relationship or attribute that applies to `Symptoms` also applies to `Fever`. The full arrows indicate relationships between concepts, e.g., a `Disease` is associated with its `Symptoms` through the `hasSymptom` relationship. The ovals represent properties of a concept, e.g., the temperature `Temp` associated with a `Fever`. Finally, smaller arrows and text represent data (individuals) associated with the concepts of the ontology, e.g., `Antibiotics` is a kind of `Treatment`. An ontology potentially also contains classification axioms, which can be used to infer new knowledge using the formally defined model. For example, the class `CureableDisease` is defined as a `Disease`, which has at least one `Treatment` associated with it through the `hasTreatment` relationship.

A semantic reasoner is a piece of generic software, which is able to infer logical consequences, i.e., new knowledge, out of the information captured in an ontology. For example, a reasoner can automatically infer which `Diseases` are `CurableDiseases`, even if this is not explicitly defined in the ontology. Three types of reasoning tasks are typically performed:

- **Checking consistency and satisfiability**: an ontology is consistent if it does not contain any contradictory statements. A class is satisfiable if it can be instantiated, i.e., an individual of this type can be created.

- **Classification** determines the complete class hierarchy of the ontology by inferring for the logically defined concepts where they belong in the existing hierarchy, defined by the subclass relationships. For example, from Figure 1.6 it can be inferred that the `CurableDisease` concept is a subclass of `Disease`.

- **Realization** calculates for each individual to which concepts it belongs.

The leading language for encoding ontologies is the Web Ontology Language (OWL) [28], which is a recommendation by the World Wide Web Consortium (W3C) [29]. OWL has different levels of expressive power. This was motivated by the principle of minimality, i.e., not including as many modelling features as possible, but constriction of expressivity to make inference feasible. It consists of three sublanguages, each of them varying in their trade-off between expressiveness and inferential complexity. They are, in order of increasing expressiveness:

- **OWL-Lite** supports classification hierarchies and simple constraint features.

- **OWL-DL (Description Logics)** is a subset providing great expressiveness without losing computational completeness and decidability.

- **OWL-Full** supports maximum expressiveness and syntactic freedom however without computational guarantees.

Using one of the three sublanguage-flavours of OWL, one can easily adapt to the required expressiveness. Arguably, the most interesting sublanguage for many application domains is OWL-DL, balancing great expressiveness with inferential efficiency. Due to its foundation in Description Logics [30], which are a family of logics that are decidable fragments of first-order logic, OWL-DL is also very flexible and computationally complete. This means that all conclusions are guaranteed to be computable. The decidability of OWL-DL, being that all conclusions will be reached in finite time, is an imported aspect as well.

Recently, a new version of OWL was proposed [31]. OWL 2 specifies three new types of sublanguages, called profiles, which have favourable computational properties and are easier to implement. The profiles are designed for increased efficiency of reasoning of specific types of applications. They are:

- **OWL 2 EL** (Existential quantification Logic), which is aimed at applications employing ontologies that contain very large numbers of properties and/or classes. It enables to perform basic reasoning tasks in time that is polynomial with respect to the size of the ontology.

- **OWL 2 QL** (Query Language), which is the ideal choice for applications that use very large volumes of instance data and where query answering is the most important reasoning task. Sound and complete conjunctive query answering can be performed in LOGSPACE with respect to the size of the data.

- **OWL RL** (Rule Language), which targets applications that require scalable reasoning without sacrificing too much expressive power. The previously discussed reasoning tasks and conjunctive query answering problems can be solved in time that is polynomial with respect to the size of the ontology.

Ontologies are more and more being picked up by the industry. A wide range of mature tools have been developed to ease the ontology construction, e.g., Protégé [32] and SWOOP [33], implementation, e.g., OWL-API [34] or Jena [35], and reasoning, e.g., Pellet [36] or Hermit [37]. To increase the expressivity, OWL can be easily integrated with different rule platforms, e.g., SWRL [38] or Jena Rules. SPARQL Protocol And RDF Query Language (SPARQL) [39] can be used to query OWL. OWL is compatible with the Web Architecture as it has, amongst others, an XML-based [40] encoding and it is backward compatible with Resource Description Framework (RDF) Schema [41].

Although originally intended for the Semantic Web [42], ontologies are becoming increasingly popular within other domains, such as knowledge representation and maintenance, biomedical informatics, context-aware computing and natural language processing.

1.3 Problem statement and research challenges

The adoption of context-aware services in the healthcare domain is lagging behind what could be expected. Whereas the healthcare industry is quick to exploit the latest medical technology, they are reluctant adopters of modern health information systems [43]. Half of all computer-based information systems fail due to user resistance and staff interference [44]. The main complaint made against mobile, context-aware systems is that users had to significantly alter workflow patterns to accommodate the system [45]. This is due to inadequate techniques for personalization of the services, a lack of focus on the soft aspects of interaction, e.g., automated and personalized alerts, and the lack of tackling problems, such as the need of the users for control [46]. To increase the adoption of context-aware healthcare service, the following research challenges need to be tackled.

A first challenge consists of **investigating how the healthcare knowledge models can be developed in a user-driven manner, without overburdening the domain experts or requiring a lot of IT knowledge**. In order to create an ontology, a good understanding of the domain it wishes to describe is required. The existing ontology engineering approaches take a rather opposed stance when it comes to including domain experts in the ontology life cycle. The first group of methodologies [47–50] emphasize the role of the knowledge engineer. Domain experts are only, mostly passively, involved at the start of the development process to discuss the scope, requirements and use of the ontology. However, including the domain expert in every step of the creation of the ontology facilitates the acceptance of the new technology, which is built using this knowledge model. Such a user-driven approach allows the domain experts to have control over the knowledge flow in their environment and adapt it to their needs. The second group of methodologies [51, 52] put the domain experts at the center of the ontology engineering process. User-friendly and collaborative tools are offered that allow them to construct, merge and discuss their own ontologies. However, this requires a considerable effort and time investment from the domain experts, as they need to acquire skills to construct the ontology themselves. This might prove a difficult task in the less IT-focussed healthcare domain where staff resources are already stretched thin. A methodology thus needs to be developed that finds the middle ground between these two approaches.

A second research challenge focuses on **building a formal and semantic model that is able to represent all the heterogeneous continuous care data, which is gathered from the various sensors, devices & healthcare databases and is communicated between the applications**. An often overlooked fact is that the strength of any context-aware platform depends heavily on the correctness and completeness of the used knowledge model. To adequately support the caregivers and care receivers and to develop healthcare services that take their needs and preference into account to optimize continuous care, this model needs to capture the daily work practices and context accurately [53]. However, most of the developed healthcare ontologies focus on biomedical research and are mainly employed to clearly define medical terminology [54], e.g., the Galen Common Reference Model [55] or Gene Ontology [56]. Little work has been done on developing high-level ontologies, which can be used to model context information and knowledge utilized across the various continuous care settings, e.g., hospitals, care residences and homecare.

An important third research challenge consists of **investigating how healthcare services, which use the knowledge model to optimize continuous care, can be developed and deployed in a scalable, modular and performant manner. This entails research on how the knowledge model can be deployed in a distributed manner and how the large amount of heterogeneous data can be dynamically filtered such that the different applications only receive the healthcare data that is relevant to them at that moment.** The amount of heterogeneous data provided by the monitoring equipment, captured in medical databases and generated by the available software is vast. Morris [57] reports over 236 different variable categories in a medical Intensive Care Unit (ICU) record. It is generally assumed that every ICU patient generates around 16,000 different data values on a daily basis. Processing all this data with a centralized knowledge component, consisting of a semantic model and accompanying algorithms, severely deteriorates the performance and scalability [58].

A fourth challenge focuses on **researching how the healthcare services to optimize continuous care can be made more self-adaptable to the future needs of the users, by automatically adjusting the knowledge model and how the context information is taken into account**. The context-aware platforms use dynamic algorithms, which take the context information into account, to adapt the behavior of the applications according to the context and offer personalized services to the users. However, these algorithms are defined at development time. When new technology is introduced, the behavior of the users changes to adapt to it. Moreover, different environments, in which the application is deployed, e.g., different nursing units or hospital departments, might have slightly different requirements pertaining to how the context information is taken into account. It is difficult to foresee these changes in behavior and small nuances in workflows at development time. This means that the context model might be incomplete or the associated algorithms may no longer apply. As the applications do not adapt to the requirements and workflow patterns of the users, they feel less in control of the technology and have to adapt their behavior to accomodate the technology instead of the other way around.

Representing medical time series accurately in the knowledge model and developing decision support tools that can discover trends and patterns in time-dependent data is a fifth important research challenge. Medical time series contain important information about the condition of a care receiver. However, it is difficult for caregivers to continuously monitor these time-dependent parameters for subtle or sometimes even overt changes that suggest a relevant clinical deterioration due to a complication or new pathology, because of the large amount of data and staff shortage. It is thus important to take this temporal data into account when developing healthcare applications and services.

Addressing these challenges will result in a pervasive, context-aware and user-centered healthcare platform that is able to optimize continuous care process in a scalable and performant manner, offer personalized healthcare services to the caregivers and care receivers, is tuned towards to needs and preferences of these users and constantly adapts itself to their changing requirements.

1.4 Main research contributions

This dissertation contributes to the research fields of health informatics, semantics and knowledge management and discovery. The main focus of this PhD research is the design and development of a context-aware, semantic and self-learning framework and accompanying methodologies and algorithms, which allow the user-driven development of pervasive healthcare applications that support caregivers and care receivers in their daily activities and tasks. This main goal translates into six research contributions. The first five map on the five research challenges discussed in the previous section. The last contribution is the development of a self-learning, ontology-based Nurse Call System (oNCS) prototype to highlight the performance, correctness and applicability of the other research contributions. The research contributions are:

1. A participatory ontology engineering methodology[1], which achieves a user-driven development process that results in ontologies and accompanying reasoning algorithms that are tuned towards the needs, requirements and daily work practices of the caregivers and care receivers.

    - A methodology to develop an ontology and accompanying reasoning constraints and domain rules, involving social scientists, ontology engineers and stakeholders, i.e., professionals working for the healthcare industry and targeted end-users, e.g., doctors, nurses and residential caregivers. The methodology consists of observations and five types of workshops, covering the whole ontology life-cycle from specification over conceptualization, formalization and implementation to maintenance. The methodology is tuned towards the development of ontologies for less IT-focused domains, where the stakeholders might not be willing or able to construct the ontologies themselves or attribute a large amount of their time.
    - Guidelines detailing how the ethnographic research and different workshop should be organized, allowing other researchers and developers to easily adopt the methodology. The guidelines stipulate for the observations and each workshop the objective, the number of required participants and their profile, the different steps used to reach this goal, the instruments used to capture, process and analyze the gained insights and how these are translated to the ontology and accompanying reasoning constraints and rules.
    - A thorough evaluation of the proposed methodology by applying it to develop a continuous care ontology.
    - An in-depth discussion and reflection on the designed methodology, resulting in nine lessons learned that should be taken into account when involving users in the ontology development process.

[1]This research was performed in collaboration with the researchers from the center for Studies on Media, Information and Telecommunication (SMIT) at Brussels University and the Centre for User Experience Research (CUO) at KU Leuven. A truly user-centered ontology engineering process was achieved by a close interaction between my technical expertise and their expertise in user research and social sciences.

2. A modular continuous care ontology, modeling context information and knowledge utilized across the various continuous care settings, i.e., hospitals, residential care and homecare.

    - The design of hierarchical ontologies for the continuous care domain, including modeling principles and mechanisms that allow to adapt the ontologies over time.
    - A high-level ontology, called the continuous care core ontology, which contains knowledge that is applicable across all continuous care domains and is of interest to a plethora of healthcare applications and services. The ontology was designed in a modular way instead of as one big semantic model, which facilitates (partial) re-use. The following seven core ontologies were developed: the *Upper*, *Sensor*, *Context*, *Profile*, *Role* & *Competence*, *Medical* and *Task* continuous care core ontologies.
    - Two low-level ontologies modeling knowledge particular to a specific continuous care domain, namely the low-level *Cure* and *Care* ontology, which are respectively tuned towards the knowledge exchanged in hospital and continuous care settings. Each of these models also consists of a number of ontologies, extending specific core ontologies.
    - A thorough evaluation of the applicability and completeness of the ontology by incorporating it into two prototypes, namely the ontology-based Nurse Call System (oNCS) and a task management system.

3. A context-aware and semantic platform, called the O'Care Platform, which allows the scalable, modular and performant deployment of healthcare applications and services.

    - A detailed description of how the Semantic Communication Bus (SCB) can be used in combination with the continuous care core ontologies to dynamically filter the large amount of heterogeneous context and health data, so that the different healthcare applications and services only receive the data that is relevant to them at that moment. The SCB was developed by Famaey et al.(SCB) [59] to accomplish a flexible and semantic publish/subscribe mechanism to communicate information between the applications and devices delivering information and the applications processing this information.
    - The O'Care Platform, which can be used to develop scalable and modular healthcare applications by enabling distributed management of the context model and information, intelligent filtering of context information, distributed deployment of the SCB and use of a cache. The plat-

forms also allows expressive OWL-DL context reasoning. The combination of all these features differentiates this platform from other works in the same area.

- A thorough performance evaluation of the SCB in the healthcare domain by an illustrative scenario concerning three healthcare applications, namely a sophisticated nurse call system integrated with localization and home automation services.

4. A self-learning framework, allowing context-aware healthcare applications to adapt their behavior at run-time to accomodate the changing requirements of the users and achieve truly personalized healthcare services.

   - A detailed description of the pipelined architecture of the self-learning framework, consisting of the following modules:
     - monitoring algorithms to identify missing or inaccurate knowledge in the context-aware healthcare applications and services,
     - configuration and data collection modules to capture the behavior of the user and the accompanying context from the ontology-based, context-aware platform,
     - pre-processing and input convertor modules to process and structure the gathered historical data,
     - data mining techniques that discover trends and patterns in this historical data,
     - a post-processing module to translate the results of the data mining techniques into knowledge that can be incorporated in the healthcare applications,
     - a decision module, consisting of algorithms to filter and prioritize this knowledge by associating probabilities that express how reliable or accurate it is, and
     - an integration module to incorporate this probabilistic knowledge into the context model and accompanying dynamic algorithms of the healthcare applications.
   - Algorithms to adapt, i.e., in- or decrease, the probabilities associated with the learned knowledge according to contex and behavior information gathered about the usage of this information. This allows less relevant knowledge to slowly disappear to the background of the healthcare applications, while trusted knowledge becomes an integral part of them.
   - A thorough evaluation of the applicability, correctness and performance of the self-learning framework in the healthcare domain by an illustra-

tive scenario, concerning the discovery of reasons for patients' call light use.

5. A prototype of a self-learning oNCS, thoroughly demonstrating how intelligent and performant healthcare services can be implemented to optimize continuous care using the continuous care ontology, the O'Care Platform and the self-learning framework.

    - Investigating methods to represent and reason with probabilistic information in ontologies and evaluate the performance of the chosen method, i.e., Pronto [60].
    - Description of the mechanisms to incorporate ontology models in healthcare applications and how they can be used to develop dynamic algorithms by presenting a dynamic nurse call algorithm, consisting of:
        - an algorithm to determine the probability that a call has a particular priority, based on the type of the call and the risk factors of the patient,
        - a threshold algorithm to determine the priority of a particular call based on the probabilistic information in the ontology,
        - a nurse call algorithm, which assigns the most appropriate caregiver to a call based on the available context information and the priority of the call.
    - Elaborate evaluation of applicability and performance of the developed oNCS prototype. First, intricate simulations were performed, based on data gathered about three departments at Ghent University Hospital. Second, user tests were conducted in the Patient Room of the Future (PRoF), using the developed mobile nurse call application to give the invited caregivers a first-hand experience of the dynamic nurse call system. The arrival times of nurses at the location of a call, the workload distribution of calls amongst nurses and the assignment of priorities to calls are compared for the oNCS and the current, place-oriented nurse call system. Additionally, the performance of the nurse call algorithm is presented and the sensitivity of the threshold algorithms is explored.
    - Algorithms to automatically adapt the parameters of the oNCS, i.e., the defined probabilities and thresholds, to the needs and requirements of caregivers and care receivers of the specific department where the oNCS is deployed.

6. An extension of the developed O'Care Platform, continuous care ontology and self-learning framework to represent and reason with medical time series.

- Investigate techniques to represent time-dependent data in ontologies and extend the continuous care ontology with relations and concepts to model temporal, medical knowledge.
- Evaluating machine learning techniques to discover trends in time series with a medically relevant use case scenario, namely determining whether a patient has sepsis. Sepsis [61] is a severe inflammatory response of the body to an infection.
- Detailed evaluation of the advantages of using echo state networks (ESN) instead of other traditional classifiers, combined with feature extraction and selection, for classification problems in the ICU when the input data consists of time series. ESN is used to predict the need for dialysis between the fifth and tenth day after admission in the ICU. The classification performance and computation time needed to preprocess the data and train and test the classifier of the ESN is compared to support vector machines and the naive Bayes classifier.

1.5 Outline of this dissertation

This dissertation is composed of a selected number of publications, which were written in the context of this PhD research. Together, they present an integral and consistent overview of the work performed. They offer significant contributions to scientific research related to the design and management of ontology-based platforms for the realization of healthcare applications and services. This section provides an overview of the remainder of this disseration and explains how the different chapters are linked together. Figure 1.7 presents the iterative methodological approach used in this research, while Figure 1.8 positions the different contributions that are presented in the next chapters and appendices. It can be noted that an iterative approach was adopted in this research, in which the developed methodologies, platforms and algorithms were fine-tuned by incorporating feedback from the implemented use cases. Given the diversity of the research domains touched upon in this dissertation, a detailed of overview of the state of the art is presented in each chapter pertaining to the research topics discussed in that chapter.

Chapter 2 presents the participatory ontology engineering methodology, as described above in Contribution 2. It starts with an overview of the existing ontology engineering methodologies and eHealth ontologies and highlights their shortcomings. Next, a series of workshops is described that were developed and organized to actively involve domain experts in the ontology engineering process. The continuous care ontology, which was highlighted in Contribution 1 and resulted from applying the methodology, is also discussed in detail. Finally, Chapter 2 ends with a discussion of the most important lessons learned during the co-creation of

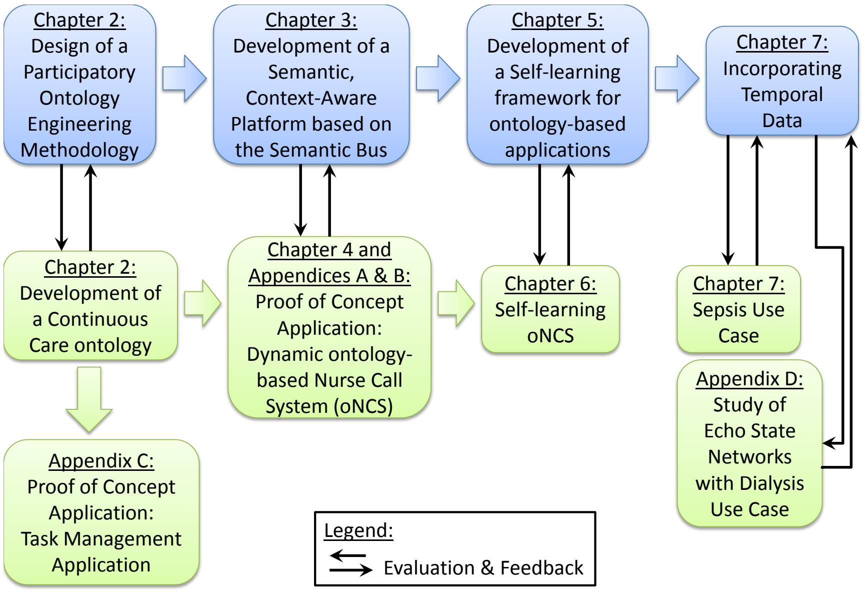


Figure 1.7: Workflow of the iterative methodological research approach

the ontology with the participatory methodology. To achieve a deeper reflection on the concepts and axioms captured in the ontology in workshop 5, a prototype application was developed, which uses the ontology to optimize continuous care processes. The protoype, which was chosen in consultation with the different targeted end-users and stakeholders, is a dynamic nurse call or care request system, called the ontology-based Nurse Call System (oNCS). The general architecture of this prototype is discussed in Section 3.3.3 of Chapter 3, while the implementation details of the oNCS itself are highlighted in Chapter 4 and Appendix B. Finally, Appendix A presents the mobile nurse call application, which was used to give the stakeholders and users a first-hand experience of the oNCS in order in order to generate deeper reflection on both the application and the ontology, on which the application was built. To further demonstrate the general applicability of the developed ontology, a task management application was developed, which intelligently assigns priorities and caregivers to tasks based on the continuous care context information captured in the semantic model. This task management application is described in detail in Appendix C.

Chapter 3 details the O'Care Platform, as discussed above in Contribution 3. It starts with an elaborate discussion of the state of the art pertaining to context-aware systems and their application to the healthcare domain. The contributions of the O'Care Platform in view of this related work are also highlighted. Next, the general architecture of the O'Care Platform is presented. The platform gathers the

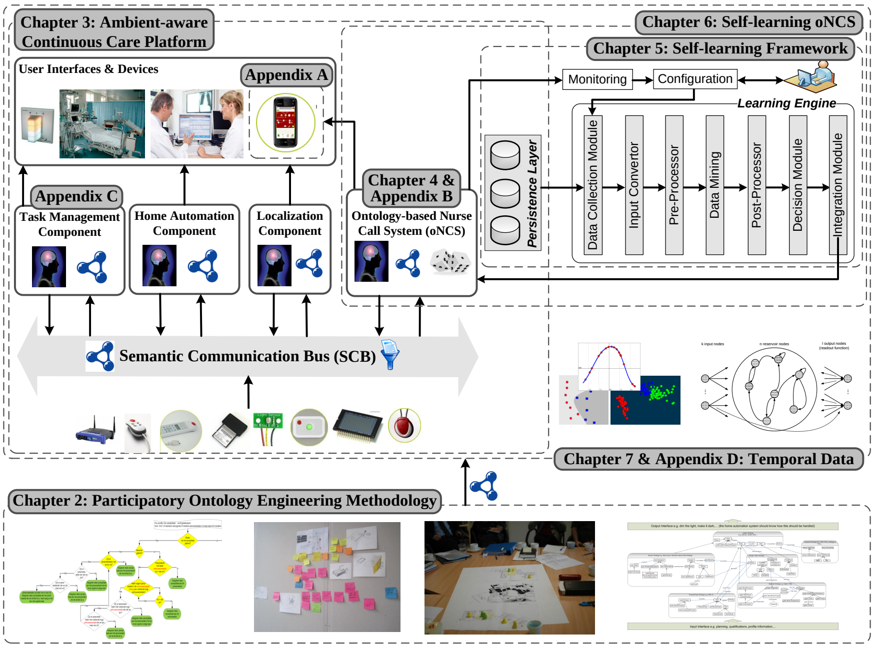


Figure 1.8: Schematic overview of the contributions of this dissertation and the chapters and appendices in which they are discussed

heterogeneous continuous care data generated by the various sensors and applications and provides it to the intelligent applications that process this data. To realize this, the O'Care Platform employs the SCB and the continuous care core ontologies, discussed in Chapter 2, to filter the large amount of heterogeneous context and health data such that the different healthcare applications and services only receive the data that is relevant to them at that moment. The intelligent healthcare applications use (a subset of) of the low-level continuous care ontologies discussed in Chapter 2 to model their specific (sub)domain and perform sophisticated reasoning. Note that the SCB does not retain any data. The gathered knowledge is distributed across the various applications and the SCB is used as communication substrate. As such a truly scalable and modular platform is achieved. It is also discussed how the scalability of the O'Care Platform can be improved by extending the SCB with a cache and deploying the SCB in a distributed fashion. Finally, it is shown how the intelligent applications can automatically subscribe to new types of data based on the context information they receive. This chapter elaborates on the specifics of the platform using an illustrative example, namely realizing the oNCS supported by a *Localization* and *Home Automation Component*. As mentioned previously, the oNCS is further discussed in Chapter 4 and Appendices B and A. To test and demonstrate the advantages and performance of the O'Care Platform, the prototype was evaluated in a living lab environment. The amount of data that is filtered by the O'Care Platform and the performance and scalability of the filter rules are evaluated for the healthcare scenario under scrutiny.

As mentioned previously, Chapter 4 details the oNCS as an illustration of how intelligent healthcare applications and services can be written to optimize continuous care, using dynamic algorithms based on the context data gathered in the continuous care ontology. This chapter focusses on the application as a whole by presenting a) its general architecture, b) how the profiles of the staff members and patients are managed using the continuous care ontology, c) the novel nurse algorithm that takes the provided context information into account to assign the most suitable caregiver to a call, and d) the implementation details. Moreover, simulation results are presented based on data gathered about one department at Ghent University Hospital. Finally, the performance of the novel nurse call algorithm is evaluated. In contrast, Appendix B focusses more on a detailed description of the probabilistic algorithms used to determine the probability that a call has a specific priority and the threshold algorithm used to determine the most suitable priority for a call based on these probabilities. This appendix also discusses the different existing approaches for respresenting and reasoning with probabilistic information in ontologies, motivates the choice for Pronto and evaluates its performance. Moreover, data was gathered about two additional departments at Ghent University Hospital. Their simulation results are discussed and compared to each other and to the results of the first department. Finally, the sensitiviy of the threshold algorithm is explored. In both papers, the simulations compare the arrival times of nurses at the location of a call, the workload distribution of calls amongst nurses and the assignment of priorities to calls between the oNCS and the current nurse call system. As mentioned earlier, an iterative methodology was employed in this research. As a result of the feedback obtained during the evaluation of the prototype with the users as part of the participatory ontology engineering methodology, the nurse call algorithm was updated. The new version of the nurse call algorithm is detailed in Appendix A. This appendix also contains a description of the mobile nurse call application and summarizes the ten most important lessons learned pertaining to the development of a dynamic nurse call system. These two appendices and this chapter thus capture the research contributions highlighted in Contribution 5.

Chapter 5 presents the self-learning framework, which allows context-aware healthcare applications to adapt their behavior at run-time. First, the relevant related work on self-learning context-aware systems is presented. Next, the general architecture of the self-learning framework is described, consisting of the different modules discussed in Contribution 4 of the previous section. The implementation of this generic framework is discussed and its applicability is illustrated by an in-depth discussion of the implementation of a specific use case, namely mining the reasons for patients' call light use. It is shown how historical information about the context and behavior of users can be gathered with the continuous care ontologies, presented in Chapter 2. Next, it is discussed how new knowledge can be discovered by mining this historical data with decision trees and translating the results to rules. Algorithms are presented to filter and prioritize the rules by associating probabilities that express the accuracy of the discovered knowledge. It is also shown how this new knowledge and associated probabilities can be integrated into the ontology and accompanying algorithms. Finally, it is discussed how these probabilities can be adapted as more information comes available that confirms or negates the new knowledge. The correctness of the discovered knowledge is evaluated in function of the size of and the amount of noise in the data set. The execution time and memory usage of the self-learning framework is also discussed. Chapter 6 details how the self-learning framework can be used to automatically adapt the parameters of the oNCS, i.e., the defined probabilities and thresholds, to the needs and requirements of the specific department where the oNCS is deployed.

Finally, Chapter 7 presents an extension of the developed OCare Platform (Chapter 3), continuous care ontology (Chapter 2) and self-learning framework (Chapter 5) to respresent and reason with medical time series, as discussed in Contribution 6. First, existing techniques to represent and reason with temporal data in ontologies are exploited to model medical time series in the continuous care ontology. Second, a specific use case is employed, namely diagnosing whether a patient has sepsis, to investigate specific machine learning techniques to classify time series data. Third, it is discussed how the results of this classification, i.e., patient has sepsis or not, can be integrated in the ontology with an associated probability expressing its reliability. Appendix D dives further into the investigation of machine learning techniques to classify time series data by presenting an elaborate evaluation of the advantages of using ESN instead of other traditional classifiers, combined with feature extraction and selection. The usefulness of ESN is shown by using it to predict the need for dialysis between the fifth and tenth day after admission in ICU patients. The performance results and pre-processing, train and test time of the ESN are compared to the performance and execution times acquired by using Support Vector Machines (SVMs) and the Naive Bayes (NB) classifier combined with feature extraction and selection. A hybrid filter-wrapper feature selection method is used with an NB classifier as classifier.

1.6 Publications

The research results obtained during this PhD research have led to several publications in peer reviewed scientific journals and proceedings of both national and international conferences. A complete list is provided below.

1.6.1 A1: Publications in international journals indexed by the ISI Web of Science "Science Citation Index Expanded"[2]

1. **Femke Ongenae**, Matthias Strobbe, Jan Hollez, Gregory De Jans, Filip De Turck, Tom Dhaene, Piet Demeester, and Piet Verhoeve. *Design of a Semantic Person-Oriented Nurse Call Management System.* Published in International Journal of Web and Grid Services, Volume 4, Issue 3, Pages 267-283, January 2008. doi: 10.1504/IJWGS.2008.021493.

2. **Femke Ongenae**, Femke De Backere, Kristof Steurbaut, Kirsten Colpaert, Wannes Kerckhove, Johan Decruyenaere, and Filip De Turck. *Towards Computerizing Intensive Care Sedation Guidelines: Design of a Rule-Based Architecture for Automated Execution of Clinical Guidelines.* Published in BMC Medical Informatics and Decision Making, Volume 10, Issue 3, Pages 22, January 2010. doi: 10.1186/1472-6947-10-3.

3. **Femke Ongenae**, Dries Myny, Tom Dhaene, Tom Defloor, Dirk Van Goubergen, Piet Verhoeve, Johan Decruyenaere, and Filip De Turck. *An Ontology-Based Nurse Call Management System (oNCS) with Probabilistic Priority Assessment.* Published in BMC Health Services Research, Volume 11, Issue 1, Pages 26, February 2011. doi: 10.1186/1472-6963-11-26.

4. Stijn Verstichel, **Femke Ongenae**, Bruno Volckaert, Filip De Turck, Bart Dhoedt, Tom Dhaene, and Piet Demeester. *An Autonomous Service-Platform to Support Distributed Ontology-Based Context-Aware Agents.* Published in Expert Systems, Volume 28, Issue 5, Pages 437–460, November 2011. doi: 10.1111/j.1468-0394.2011.00587.x.

5. Stijn Verstichel, **Femke Ongenae**, Leanneke Loeve, Frederik Vermeulen, Pieter Dings, Bart Dhoedt, Tom Dhaene, and Filip De Turck. *Efficient Data Integration in the Railway Domain Through an Ontology-Based Methodology.* Published in Transportation Research Part C - Emerging Technologies, Volume 19, Issue 4, Pages 617–643, November 2011. doi: 10.1016/j.trc.2010.10.003.

6. Matthias Strobbe, Olivier Van Laere, **Femke Ongenae**, Samuel Dauwe, Bart Dhoedt, Tom Dhaene, Filip De Turck, Piet Demeester, and Kris Luyten. *Novel Applications Integrate Location and Context Information.* Published in IEEE Pervasive Computing, Volume 11, Issue 2, Pages 64–73, February 2012. doi: 10.1109/MPRV.2011.60.

[2]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

7. **Femke Ongenae**, Stijn Van Looy, David Verstraeten, Thierry Verplancke, Dominique Benoit, Filip De Turck, Tom Dhaene, Benjamin Schrauwen, and Johan Decruyenaere. *Time Series Classification for the Prediction of Dialysis in Critically Ill Patients Using Echo State Networks.* Published in Engineering Applications of Artificial Intelligence, Volume 26, Issue 3, Pages 984-996, March 2013. doi: 10.1016/j.engappai.2012.09.019.

8. **Femke Ongenae**, Dries Myny, Tom Dhaene, Tom Defloor, Dirk Van Goubergen, Piet Verhoeve, Johan Decruyenaere, and Filip De Turck. *Probabilistic Priority Assessment of Nurse Calls* Revisions Submitted to Medical Decision Making, March 2013.

9. **Femke Ongenae**, Jeroen Famaey, Stijn Verstichel, Saar De Zutter, Steven Latré, Ann Ackaert, Piet Verhoeve, and Filip De Turck. *Ambient-Aware Continuous Care Through Semantic Context Dissemination.* Revisions Submitted to BMC Medical Informatics & Decision Making, April 2013.

10. **Femke Ongenae**, Pieter Duysburgh, Nicky Sulmon, Matthijs Verstraete, Lizzy Bleumers, Saar De Zutter, Stijn Verstichel, Ann Ackaert, Filip De Turck, and An Jacobs. *Towards an Ontology Co-design Methodology: The Co-creation of a Continuous Care Ontology* Submitted to Applied Ontology, April 2013.

11. Femke De Backere, **Femke Ongenae**, Frederic Vannieuwenborg, Jan Van Ooteghem, Pieter Duysburgh, Arne Jansen, Jeroen Hoebeke, Kim Wuyts, Jen Rossey, Floris Van den Abeele, Karen Willems, Jasmien Decancq, Jan Henk Annema, Nicky Sulmon, Dimitri Van Landuyt, Stijn Verstichel, Pieter Crombez, Ann Ackaert, Dirk De Grooff, An Jacobs, and Filip De Turck. *Organizing Care through trusted Cloud Services: the OCareCloudS project* Submitted to Informatics for Health and Social Care, May 2013.

12. **Femke Ongenae**, Maxim Claeys, Thomas Dupont, Wannes Kerckhove, Piet Verhoeve, Tom Dhaene, and Filip De Turck. *A Probabilistic Ontology-based Platform for Self-adaptation of Context-aware Healthcare Applications* Accepted for Expert Systems with Applications, July 2013.

13. **Femke Ongenae**, Maxim Claeys, Wannes Kerckhove, Thomas Dupont, Piet Verhoeve, and Filip De Turck. *A Self-learning Nurse Call System* Submitted to Computers in Biology and Medicine, July 2013.

14. **Femke Ongenae**, Thomas Vanhove, Femke De Backere, and Filip De Turck. *Intelligent Task Management Platform for Healthcare Workers* Submitted to Journal of Medical Systems, July 2013.

1.6.2 P1: Conference proceedings indexed by ISI Web of Science “Conference Proceedings Citation Index - Science”[3]

1. **Femke Ongenae**, Matthias Strobbe, Jan Hollez, Gregory De Jans, Filip De Turck, Tom Dhaene, Piet Demeester, and Piet Verhoeve. *Ontology-Based and Context-Aware Hospital Nurse Call Optimization.* In proceedings of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Catalonia, Spain, Pages 985-990, March 2008. doi: 10.1109/CISIS.2008.80.

2. **Femke Ongenae**, Thomas Dupont, Wannes Kerckhove, Wouter Haerick, Kristof Taveirne, Filip De Turck, and Johan Decruyenaere. *Design of ICU Medical Decision Support Applications by Integrating Service Oriented Applications with a Rule-Based System.* In proceedings of the 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), Bratislava, Slovak Republic, Pages 12-17, November 2009. doi: 10.1109/ISABEL.2009.5373663.

1.6.3 C1: Other international conference proceedings

1. **Femke Ongenae**, Stijn Verstichel, Filip De Turck, Tom Dhaene, Bart Dhoedt, and Piet Demeester. *OTAGen: A Tunable Ontology Generator for Benchmarking Ontology-Based Agent Collaboration.* In proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC), Turku, Finland, Pages 529–530, July 2008. doi: 10.1109/COMPSAC.2008.134.

2. **Femke Ongenae**, Ann Ackaert, Filip De Turck, An Jacobs, Annelies Veys, Piet Verhoeve, and Mieke Van Gils. *User-Driven Design of an Ontology-Based Ambient-Aware Continuous Care Platform.* In proceedings of the 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Munich, Germany, Pages 1–4, March 2010. doi: 10.4108/ICST.PERVASIVEHEALTH2010.8907.

3. **Femke Ongenae**, Tom Dhaene, Filip De Turck, Dominique Benoit, and Johan Decruyenaere. *Design of a Probabilistic Ontology-Based Clinical Decision Support System for Classifying Temporal Patterns in the ICU: A Sepsis Case Study.* In proceedings of the 23rd IEEE International Symposium on

[3] The publications listed are recognized as ‘P1 publications’, according to the following definition used by Ghent University: P1 publications are proceedings listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

Computer-Based Medical Systems (CBMS), Perth, Australia, Pages 389–394, October 2010. doi: 10.1109/CBMS.2010.6042676.

4. Lizzy Bleumers, Nicky Sulmon, **Femke Ongenae**, An Jacobs, Mathijs Verstraete, Mieke Van Gils, Ann Ackaert, and Saar De Zutter. *Towards Ontology Co-Creation in Institutionalized Care Setings.* In proceedings of the 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Dublin, Ireland, Pages 559–562, May 2011.

5. **Femke Ongenae**, Lizzy Bleumers, Nicky Sulmon, Mathijs Verstraete, Mieke Van Gils, An Jacobs, Saar De Zutter, Piet Verhoeve, Ann Ackaert, and Filip De Turck. *Participatory Design of a Continuous Care Ontology: Towards a User-Driven Ontology Engineering Methodology.* In proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD), Paris, France, Pages 81–90, October 2011.

6. **Femke Ongenae**, Pieter Duysburgh, Mathijs Verstraete, Nicky Sulmon, Lizzy Bleumers, An Jacobs, Ann Ackaert, Saar De Zutter, Stijn Verstichel, and Filip De Turck. *User-Driven Design of a Context-Aware Application: An Ambient-Intelligent Nurse Call System.* In proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), San Diego, CA, USA, Pages 205–210, May 2012. doi: 10.4108/icst.pervasivehealth.2012.248699.

7. Niels Bouten, Anna Hristoskova, **Femke Ongenae**, Jelle Nelis, and Filip De Turck. *Ontology-Driven Dynamic Discovery and Distributed Coordination of a Robot Swarm.* In proceedings of the 6th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security (AIMS), Luxembourg, Luxembourg, Pages 2–13, June 2012. doi: 10.1007/978-3-642-30633-4_2.

8. **Femke Ongenae**, and Filip De Turck. *User-Driven Design of Ontology-Based, Context-Aware and Self-Learning Continuous Care Applications.* In proceedings of the 8th International Conference on Network and Service Management (CNSM), Las Vegas, Nevada, USA, Pages 266–270, October 2012.

9. **Femke Ongenae**, Anna Hristoskova, Elena Tsiporkova, Tom Tourwé, and Filip De Turck. *Semantic Reasoning for Intelligent Emergency Response Applications.* In proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Baden-Baden, Germany, 12-15 May 2013.

10. Anna Hristoskova, **Femke Ongenae**, and Filip De Turck. *Semantic Reasoning for Intelligent Emergency Response Applications.* In proceedings of

the 11th IEEE International Conference on Industrial Informatics (INDIN), Bochum, Germany, 29-31 July 2013.

11. **Femke Ongenae**, Stijn Verstichel, Maarten Wijnants and Filip De Turck. *A Distributed Reasoning Platform to Preserve Energy in Wireless Sensor Networks.* Submitted to the 12th International Semantic Web Conference (ISWC), Sydney, Australia, 21-25 October 2013.

1.6.4 Patents

1. **Femke Ongenae**, Filip De Turck, Tom Dhaene, Ann Ackaert, Piet Verhoeve, Brecht Stubbe. *Method and device for handling calls in a nurse call system.* European Patent Application EP2385476, September 2011.

1.6.5 Other publications

1. **Femke Ongenae**, Filip De Turck, and Tom Dhaene. *Efficient management of distributed and dynamic ontologies.* In proceedings of the 10th FirW PhD Symposium, Ghent, Belgium, Pages 90–91, December 2009.

References

[1] World Health Organization (WHO). *Health systems*. http://www.who.int/topics/health_systems/en/, 2013.

[2] Eurostat. *Population structure and ageing*. Technical report, European Commission, April 19 2013. http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Population_structure_and_ageing.

[3] World Health Organization (WHO). *Interesting facts about ageing*. http://www.who.int/ageing/about/facts/en/index.html, March 28 2012.

[4] I. Meyer, S. Müller, L. Kubitschke1, A. Dobrev, R. Hammerschmidt, W. B. Korte1, T. Hüsing, T. van Kleef, S. Otto, J. Heywood, and M. Wrede. *eCare as a way of coping with an ageing population today and tomorrow. The eCare Benchmarking study*. Technical report, European Commission, Directorate General Information Society and Media, Brussels, April 12 2013. http://ec.europa.eu/information_society/newsroom/cf/itemdetail.cfm?item_id=10182.

[5] World Health Organization (WHO). *The World Health Report 2006 - working together for health*. http://www.who.int/whr/2006/en/, 2006.

[6] American Association for Clinical Chemistry. *Rates of Chronic Disease Expected to Rise Sharply*. Clinical Laboratory News, 35(7):1, 2009.

[7] The Organisation for Economic Co-operation and Development (OECD). *Health at a Glance: Europe 2012*. Technical report, OECD Publishing, 2012. http://dx.doi.org/10.1787/9789264183896-en.

[8] J. Smith and K. Walshe. *Introduction: the current and future challenges of healthcare management*, pages 1–10. Open University Press, 2006.

[9] E. Percy. *Healthcare Challenges and Trends*. Technical report, Logica, 2012.

[10] C. Orwat, A. Graefe, and T. Faulwasser. *Towards pervasive computing in health care - A literature review*. BMC Medical Informatics and Decsion Making, 8(26):18, 2008.

[11] C. Pagliari, D. Sloan, P. Gregor, F. Sullivan, D. Detmer, J. P. Kahan, W. Oortwijn, and S. MacGillivray. *What is eHealth (4): a scoping exercise to map the field*. Journal of Medical Internet Research, 7(1):e9, 2005.

[12] D. Silber. *The case for eHealth*. In Proceedings of the European Commission's first high-level conference on eHealth, page 32, May 22-23 2003.

[13] J. Li, A. Talaei-Khoei, H. Seale, P. Ray, and C. R. MacIntyre. *Health Care Provider Adoption of eHealth: Systematic Literature Review*. Interactive Journal of Medical Research, 2(1):e7, 2013.

[14] Rand Corporation. *Health Information Technology: Can HIT Lower Costs and Improve Quality?* http://www.rand.org/pubs/research_briefs/RB9136/index1.html, 2005.

[15] M. Tentori, D. Segura, and J. Favela. *chapter VIII: Monitoring hospital patients using ambient displays*. Medical Information Science Reference, USA, 2009.

[16] J.-C. Burgelman and Y. Punie. *Close encounters of a different kind: ambient intelligence in Europe*, pages 19–35. Springer-Verlag, Berlin-Heidelberg, 2006.

[17] Y. Punie. *The future of ambient intelligence in Europe: the need for more everyday life*. Comm Strat, 57:141–165, 2005.

[18] M. Satyanarayanan. *Pervasive Computing: Vision and Challenges*. IEEE Personal Communications, 8(4):10–17, 2001.

[19] H. Byun and K. Cheverst. *Utilizing context history to provide dynamic adaptations*. Applied Artificial Intelligence, 18(6):533–548, 2004.

[20] A. K. Dey and G. D. Abowd. *Towards a better understanding of context and context-awareness*. In D. R. Morse and A. K. Dey, editors, Proceedings of the CHI Workskhop on the What, Who, Where, When and How of Context-Awareness, The Hague, The Netherlands, 1-6 April 2000. New York, NY, USA: ACM Press.

[21] T. Gu, H. K. Pung, and D. Q. Zhang. *A service-oriented middleware for building context-aware services*. Journal of Network and Computer Applications (JNCA), 28(1):1–18, 2005.

[22] H. Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, University of Maryland, Baltimore County, 2004.

[23] L. O. B. S. Santos, R. P. Wijnen, and P. Vink. *A service-oriented middleware for context-aware applications*. In Proceedings of the 5th International Workshop on Middleware for Pervasive and Ad hoc Computing: 26-30 November 2007; Newport Beach, Orange County, CA, USA, pages 37–42, Newport Beach, Orange County, CA, USA, 26-30 November 2007. New York, NY, USA: ACM Press.

[24] N. Bricon-Souf and C. R. Newman. *Context awareness in health care: A review*. International Journal of Medical Informatics, 76(1):2–12, 2007.

[25] U. Varshney. *Chapter 11: Context-awareness in Healthcare*. In Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring, pages 231–257. Springer Science + Business Media, LLC;, New York, NY, USA, 1st edition, 2009.

[26] A. Valls, K. Gibert, D. Sánchez, , and M. Bateta. *Using ontologies for structuring organizational knowledge in Home Care assistance*. Internation Journal of Medical Informatics, 79(5):370–387, 2010.

[27] T. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199–220, 1993.

[28] D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language Overview*. Technical report, World Wide Web Consortium (W3C), February 10 2004. http://www.w3.org/TR/owl-features/.

[29] *W3C: World Wide Web Consortium*. http://www.w3.org/, 2013.

[30] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[31] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. *OWL 2 Web Ontology Language Profiles*. Technical report, World Wide Web Consortium (W3C), 27 October 2009. http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/#Feature_Overview.

[32] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. In S. A. McIlraith, D. Plexousakis, and F. v. Harmelen, editors, Proceedings of the 3rd International Semantic Web Conference (ISWC), volume 3298/2004 of *Lecture Notes in Computer Science*, pages 229–243, Hiroshima, Japan, November 7–11 2004. Springer Berlin/Heidelberg.

[33] A. Kalyanpur, B. Parsia, E. Sirin, B. C. Grau, and J. Hendler. *Swoop: A Web Ontology Editing Browser*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 4(2):144–153, 2006.

[34] M. Horridge and S. Bechhofer. *The OWL API: A Java API for OWL Ontologies*. Semantic Web Journal, Special Issue on Semantic Web Tools and Systems, 2(1):11–21, 2011.

[35] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. *Jena: implementing the semantic web recommendations*. In Proceedings of the 13th international conference on World Wide Web, Alternate track papers & posters (WWW Alt.), pages 74–83, New York, NY, USA, May 17-22 2004. ACM.

[36] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL reasoner*. Journal of Web Semantics, 5(2):51–53, 2007.

[37] B. Motik, R. Shearer, and I. Horrocks. *Hypertableau Reasoning for Description Logics*. Journal of Artificial Intelligence Research, 36:165–228, 2009.

[38] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Technical report, World Wide Web Consortium, May 21 2004. http://www.w3.org/Submission/SWRL/.

[39] E. Prud'hommeaux and A. Seaborne. *SPARQL Query Language for RDF*. Technical report, World Wide Web Consortium, January 15 2008. http://www.w3.org/TR/rdf-sparql-query/.

[40] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. *eXtensible Markup Language (XML) 1.0 (Fourth Edition)*. Technical report, World Wide Web Consortium (W3C), August 16 2006. http://www.w3.org/TR/2006/REC-xml-20060816/.

[41] R. V. Guha, D. Brickley, and B. McBride. *RDF Vocabulary Description Language 1.0: RDF Schema*. Technical report, World Wide Web Consortium (W3C), 10 February 2004. http://www.w3.org/TR/rdf-schema/.

[42] T. Berners-Lee, J. Hendler, and O. Lassila. *The Semantic Web*. Scientific American, pages 29–37, 2001.

[43] T. Chin. *Technology Valued, but Implementing it into Practice is Slow* [online]. 2004. http://www.ama-assn.org/amednews/2004/01/19/bisb0119.htm.

[44] J. Anderston and C. Aydin. *Evaluating the Impact of Health Care Information Systems*. International Journal of Technology Assessment in Health Care, 13(2):380–393, 1997.

[45] J. H. Jahnke, Y. Bychkov, D. Dahlem, and L. Kawasme. *Context-aware information services for health care*. In T. Roth-Berghofer and S. Schulz, editors, Proceedings of the 27th German Conference on Artificial Intelligence, Workshop on Modeling and Retrieval of Context (MRC): 20-21 September 2004; Ulm, Germany, pages 73–84. Aachen, Germany: CEUR;, 2004.

[46] J. Criel and L. Claeys. *A transdisciplinary study design on context aware applications and environments. A critical view on user participation within calm computing*. Observatorio (OBS*), 2(2):57–77, 2008.

[47] M. Grüninger and M. Fox. *Methodology for the design and evaluation of ontologies*. In International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.

[48] M. Uschold and M. King. *Towards a methodology for building ontologies*. In International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.

[49] Y. Sure, S. Staab, and R. Studer. *Chapter on Ontology Engineering Methodology*, pages 135–152. Springer, Berlin-Heidelberg, 2009.

[50] M. Fernández, A. Gómez-Pérez, and N. Juristo. *METHONTOLOGY: from ontological art towards ontological engineering*. In American Asociation for Artificial Intelligence (AAAI) Spring Symposium Series on Ontological Engineering, pages 33–40, Stanford, USA, 1997.

[51] K. Kotis and G. A. Vouros. *Human-centered ontology engineering: The HCOME methodology*. International Journal of Knowledge and Information Systems (KAIS), 10:109–131, 2006.

[52] P. Spyns, Y. Tang, and R. Meersman. *An ontology engineering methodology for DOGMA*. Applied Ontology, 3(1–2):13–39, 2008.

[53] J. Ash, D. Sittig, K. Guappone, R. Dykstra, J. Richardson, A. Wright, J. Carpenter, C. McMullen, M. Shapiro, A. Bunce, and B. Middleton. *Recommended practices for computerized clinical decision support and knowledge management in community settings: a qualitative study*. BMC Medical Informatics and Decision Making, 12(1):6, 2012.

[54] F. Ongenae, F. De Backere, K. Steurbaut, K. Colpaert, W. Kerckhove, J. Decruyenaere, and F. De Turck. *Appendix B: overview of the existing medical and natural language ontologies which can be used to support the translation process*. BMC Medical Informatics and Decision Making, 10(3):4, 2011.

[55] A. L. Rector, J. E. Rogers, P. E. Zanstra, and E. van der Haring. *OpenGALEN: Open Source Medical Terminology and Tools*. In Proceedings of the annual American Medical Informatics Association (AMIA) Symposium: 8-12 November 2003; Washington, DC, USA, page 982, Washington, DC, USA, 8-12 November 2003. American Medical Informatics Association (AMIA);. http://www.opengalen.org/.

[56] J. A. Blake and M. A. Harris. *The Gene Ontology (GO) project: structured vocabularies for molecular biology and their application to genome and expression analysis*. Current Protocols in Bioinformatics, 23(7.2.1-7.2.9):1472–6947, 2008. http://www.geneontology.org/.

[57] A. H. Morris. *Computerized protocols and bedside decision support*. Critical Care Clinics, 15(3):523–545, 1999.

[58] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan, and S. Liu. *Towards a Complete OWL Ontology Benchmark*. In Y. Sure and J. Domingue, editors, Proceedings of the 3rd European Semantic Web Conference, pages 125–139, Budva, Montenegro, 11-14 June 2006. Berlin/Heidelberg, Germany: Springer;.

[59] J. Famaey, S. Latré, J. Strassner, and F. De Turck. *Semantic Context Dissemination and Service Matchmaking in Future Network Management*. International Journal of Network Management, 22(4):285–310, 2011.

[60] P. Klinov. *Pronto: A Non-monotonic Probabilistic Description Logic Reasoner*. In S. Bechhofer, M. Hauswirth, J. Hoffman, and M. Koubarakis, editors, Proceedings of the 5th European Semantic Web Conference (ESWC), pages 822–826, Tenerife, Canary Islands, Spain, June 1-5 2008. Berlin: Springer. Available from: http://pellet.owldl.com/pronto.

[61] J. F. Dhainaut, A. F. Shorr, W. L. Macias, M. J. Kollef, M. Levi, and K. R. en D. R. Nelson. *Dynamic evolution of coagulopathy in the first day of severe sepsis: relationship with mortality and organ failure*. Critical Care Medicine, 33(7):450–452, 2005.

2

An Ontology Co-design Methodology for the Co-creation of a Continuous Care Ontology

F. Ongenae, P. Duysburgh, N. Sulmon, M. Verstraete, L. Bleumers, S. De Zutter, S. Verstichel, A. Ackaert, A. Jacobs, and F. De Turck.

Submitted to Applied Ontology, April 2013.

"The most underutilized resource in all of healthcare is the patient.."

– Dave deBronkart, Meet e-Patient Dave (2011)

Context-aware techniques are not being actively adopted in healthcare to optimize continuous care processes. The main complaint is that users had to significantly alter work practices to accomodate the new technology instead of the other way around. The developed context-aware healthcare services are not really personalized towards the specific needs and users of the users. The completeness and correctness of the used knowledge model is an important factor to ensure that the context-aware applications are tuned towards the daily practices of the caregivers and care receivers. To realize such a user-centered model it is important to involve domain experts in every step of its development. However, the current methodologies either only involve the stakeholders during the specification phase

of the ontology to define its scope and requirements or require the domain experts to discuss and construct the ontologies themselves. Therefore, this chapter proposes a participatory ontology engineeringing methodology, which allows that social scientist, ontology engineers and the stakeholders co-create the ontology. It is tuned towards less IT-focussed domains where the users might not be willing or able to construct the ontology themselves or attribute a large amount of their time to this development. To evaluate the methodology a continuous care ontology was co-created. Although ontologies are an active research area within the healthcare domain, the developed ontologies focus on biomedical research. Little work has been done on developing high-level ontologies, which can be used to model context information and knowledge utilized across the various continuous care settings.
In this chapter the research is discussed that was performed to realize Research Contributions 1 and 2 highlighted in Section 1.3 of Chapter 1. To achieve a deeper reflection on the concepts and axioms captured in the ontology in workshop 5, a prototype of a dynamic nurse call system was developed. The mobile application and dynamic nurse call algorithm used to give the domain experts a first-hand experience of the prototype are detailed in Appendix A. The general architecture and set-up of the prototype are more thoroughly discussed in Section 3.3.3 of Chapter 3. While Chapter 4 and Appendix B elaborate on the dynamic nurse call system, its performance and its advantageous towards providing more personalized continuous care. To further demonstrate the general applicability of the developed ontology, Appendix C describes a task management system, which dynamically adapts the assignment of tasks based on the context.

⋆ ⋆ ⋆

Abstract
Ontology engineering methodologies tend to emphasize the role of the knowledge engineer or require a very active role of domain experts. In this paper, a participatory ontology engineering methodology is described that holds the middle ground between these two ’extremes’. After extensive ethnographic research, an interdisciplinary group of domain experts closely interacted with ontology engineers and social scientists in a series of workshops. Once a preliminary ontology was developed, a dynamic care request system was built using the ontology. Additional workshops were organized involving a broader group of domain experts to ensure the applicability of the ontology across other continuous care settings. The proposed methodology successfully actively engaged domain experts in various phases of the ontology construction, without overburdening them. Its applicability is illustrated by presenting the co-created continuous care ontology. The lessons learned during the design and execution of the methodology are also presented.

2.1 Introduction

2.1.1 Background: The ambient care room

As pervasive computing [1] continues to form one of the most exciting trends of information technology, it also poses an interesting challenge to manage and transform data from unobtrusive technology into intrusive information. Future pervasive computing applications are envisioned to adapt the applications' behavior by interpreting various contexts of an environment and its users. Such context information may often be ambiguous and heterogeneous, which makes the delivery of unambiguous context information to applications extremely challenging [2]. The growing interest in context-aware applications, which are adaptive and capable of acting autonomously on behalf of users, is spreading across domains such as education, transportation, manufacturing and healthcare [3].

When looking at healthcare, this idea of pervasive computing has led to the idea of the ambient intelligent care room of the future [4, 5]. This room contains numerous collaborative devices and as such supports both caregivers and care receivers, i.e., patients or residents, in carrying out their daily activities and tasks. All the heterogeneous data captured in the room by sensors, by wireless devices and input from caregivers are combined to meet the needs and preferences of its users. However, this vision is far from current reality. Today, caregivers mainly coordinate, manage and consult the single devices themselves. No integration is done of the several sources and devices caregivers use for consulting and entering data [6]. Due to the fact that the data available in the care room are not being integrated and aggregated, caregivers lose time, miss out on potential insights about care receivers and lack a general overview of the current situation.

For instance, consider a patient who suffers from a concussion. Such patients are sensitive to noise and light and should rest in a quiet and dark environment. It is the nurse's task to make sure the lights are dimmed when she enters and she has to inform the other staff members and visitors of the sensibilities of the patient. If the lights are switched on by anybody who is unaware of or forgets about the patient's condition, it might delay the patient's recovery. In an ambient intelligent care room, the system would know that the patient suffers from concussion and would adjust the settings of the light accordingly whenever a person enters the room. Moreover, a message could be displayed informing the visitors of the room of the patient's condition and reminding them too keep quiet and not turn on the bright lights.

The definition and use of ontology in the medical domain to tackle these issues is an active research field, as it has been recognized that ontology-based systems can be used to improve the management of complex health systems [7]. An ontology formally describes the concepts in a domain, their attributes and their relations. It can also contain classification rules. This commonly agreed upon data-format can then be used to exchange the data and its attached domain model. In this way an ontology encourages re-use, communication, collaboration and integration [8]. Examples of context-aware systems, based on ontologies, can be found in Paganelli and Giuli [9], Fook et al. [10], Zhang et al. [11] and Ko et al. [12]. Supporting caregivers and care receivers in their daily activities by developing an intelligent, context-aware framework, which exploits and integrates the available heterogeneous data by employing an ontology, was the focus of the project ACCIO (Ambient aware provisioning of Continuous Care for Intra-muros Organizations) [5, 13].

In order to create an ontology, a good understanding of the domain it wishes to describe is required. Various methods have been explored for capturing a domain by involving the people that are part of it. In the next section, two different established ontology creation methods are described. We will point out the shortcomings, and propose our own, participatory method for capturing a domain and its users for ontology creation.

2.1.2 Related Work

Ontology engineering is formally defined as [8]: “the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies”. There are five widely accepted stages for building an ontology [14]:

1. *Specification*: The purpose and the scope of the ontology are identified.
2. *Conceptualization*: A conceptual model of the ontology is constructed. It consists of the different concepts, relations and properties that can occur in the domain. This step can include studying other existing ontologies that can be reused.
3. *Formalization*: The conceptual model is translated into a formal model for example by adding axioms that restrict the possible interpretations of the model.
4. *Implementation*: The formal model is implemented in a knowledge representation language, for example the Web Ontology Language (OWL) [15].
5. *Maintenance*: The implemented ontology has to be constantly evaluated, updated and corrected. To update an ontology the previous steps can be used.

There are also activities that should be performed during the whole life cycle, namely knowledge acquisition, evaluation and documentation. Various ontology engineering methodologies have been proposed that describe methods to realize the goals of each of these stages. A comprehensive overview of the current state of the art in ontology engineering can be found in Simperl et al. [16]. The well-known methodologies can be divided into two major groups depending on the role the domain experts play.

The first group of methodologies, such as TOVE [17], ENTERPRISE [18], OTK [19] and METHONTOLOGY [20], emphasize the role of the knowledge engineer. Domain experts are only, mostly passively, involved during the specification phase to discuss the scope, requirements and use of the ontology. No methods or tools are offered to empower the domain experts to actively participate in the ontology life-cycle.

The second group, such as HCOME [21] and DOGMA [22], put the domain experts at the center of the ontology engineering process. User-friendly and collaborative tools are offered that allow them to construct, merge and discuss their own ontologies. The knowledge engineer delivers (technical) support in this process.

As can be seen, the existing approaches take a rather opposed stance when it comes to including domain experts in the ontology life cycle. Including the domain expert in the creation of the ontology facilitates the acceptance of the new technology that is built using this knowledge model, since such a user-driven approach allows the domain experts to have control over the knowledge flow in their environment and adapt it to their needs. However, while HCOME actually puts the domain experts at the center of the ontology engineering process, it also requires them to make a considerable time-investment and acquire skills to construct the ontology themselves. This is a considerable effort, one that is not always feasible for domain experts. This conclusion was also acknowledged by the survey of Simperl et al. [16], which found that current methodologies are very generic when it comes to issues of knowledge elicitation from domain experts. The assumption underlying these methodological approaches is that there exists an imperative need for a close interaction between domain experts and ontology engineers, but extensive studies on which techniques should be used for this are largely missing.

In this paper, an ontology engineering methodology is described that finds the middle ground between the two extremes. It aims at involving the domain experts in each step of the ontology life cycle without asking them to construct the ontology themselves or attribute a large amount of their time. The methodology acknowledges that domain experts are not ontology engineers and vice versa. To reach this goal, user-driven and participatory methods and tools are employed to realize each of the five stages. As such, the ontology is co-created with the domain experts. Co-creation has been described as "any act of collective creativity that is shared by two or more people" [23]. When integrated in a design process, it is also called co-design. Co-design has its roots in participatory design, an approach towards computer systems design in which the people destined to use the system also play a critical role in designing it [24]. Ontology co-creation thus refers to a creative, continuous involvement of stakeholders, i.e., (in)direct users, and ontology engineers in the ontology engineering process. The rationale behind this approach is that it increases the acceptance of ontology-driven technologies and facilitates their appropriation. It encourages users to feel in control of the ontology, continue to adapt it and to thus increase its accuracy.

Kuziemsky and Lau [25] see ontology co-creation as the key to the challenge of creating an ontology that is both accurate and useful. They observe how little research has been done in a field that nevertheless greatly benefits from high-quality and practical ontologies: (health)care. Indeed, caregivers are faced with a vast amount of information that they need to integrate and prioritize. By implementing a context-aware, ontology-based framework, certain care tasks could be automated to alleviate caregivers' workload.

Although ontologies are widely accepted within the eHealth domain, most of these models focus on biomedical research, e.g., Galen Common Reference Model [26] or the Gene Ontology [27]. Ongenae et al. [28] give an overview of the most relevant, well-known and well-developed eHealth ontologies. Little work has been done on the development of ontologies to support continuous care.

2.1.3 Definitions and terminology

The objectives of the interdisciplinary two year project ACCIO were threefold:

1. the development of a continuous care ontology,
2. the development of continuous care concepts or applications that utilize the ontology for pervasive computing purposes, and
3. the development of a methodology to involve domain experts in the creation of a continuous care ontology, without overburdening them.

These objectives are inevitably closely interrelated as the domain experts needed to be involved for the development of the ontology and determine the focus of the continuous care concepts/applications. Also, one concept was implemented as a prototype in order to validate the ontology by involving more users. Since these objectives are interdependent, we will reflect on them simultaneously throughout the description of the co-creation methodology.

Before we go deeper into the description of the methodology, we first wish to introduce a few concepts that formed the common thread throughout the process and are thus essential for understanding how the project objectives were attained. These concepts include the *high- and low-level ontologies*, the *dynamic nurse call system*, and the *innovation binder* coordination instrument.

2.1.3.1 High and low level ontologies

In the course of constructing the continuous care ontology, it became clear that it had to be subdivided in a high-level and several low-level ontologies. This has to do with the fact that the ACCIO project aimed at developing an ontology that would be applicable in both care residences (care settings) as in hospitals (cure settings). In the high-level ontology, concepts and relations that apply for both care and cure settings were described. Concepts and relations that specifically related to a care or a cure setting were included in the low-level ontology for this specific domain.

2.1.3.2 Dynamic nurse call system

To achieve a deeper reflection on the concepts and axioms captured in the ontology, one future care application was chosen to be developed as a prototype. This chosen prototype was a dynamic nurse call or care request system. The reason for this choice is motivated in Section 2.2.2.

Traditional nurse call systems are static as calls are made by buttons fixed to a wall and the nurse call algorithm consists of predefined links between rooms and caregivers' beepers [29]. They do not take into account the current situation to assist the user in making calls, assign a caregiver to a call or detect hazardous situations for which a call should be made. Moreover, the beepers give the caregivers limited context information about the call.

The dynamic, ambient-intelligent nurse call system developed in this research [30] provides each care receiver and caregiver with a badge to locate this person. Each badge also has a call button allowing care receivers and staff to walk around freely and still make (assistance) calls. When the ambient-intelligent system receives or generates a call, a decentralized algorithm for call triage finds the best-suited caregivers to answer the care receiver's request. For this, the algorithm makes use of all kinds of heterogeneous sensor data, e.g., vital parameters and light sensors, and context and user-specific parameters, e.g., locations, qualifications, occupation and trust relations, that are captured in the continuous care ontology, which was developed using the participatory ontology engineering methodology described in this paper. The novel nurse call system is also able to automatically generate new (context) calls based on this data. More information about the developed prototype can be found in Ongenae et al. [31].

2.1.3.3 Innovation binder

In order to keep an overview of the different project objectives and their interrelatedness, the so-called innovation binder was used as a central project coordination instrument. The development of the innovation binder was an iterative, evidence-based process, involving all the stakeholders and incorperating all the insights and research results obtained during the various steps of the participatory development process. The innovation binder contains a set of carefully constructed personas [32]. Personas communicate common attitudes, desires, behaviors and frustrations for a particular user group. Their main advantage is that they allow feeling real empathy for the user group, as they put a human face to a list of requirements. The innovation binder also contains a sunny-day scenario. A scenario is a short story that describes the hypothetical use of a system to help develop a detailed and shared understanding of the context and activities of the users. The scenario is sunny-day because it is an ideal scenario in which the technology, e.g., the context-aware system, the sensors and actuators, optimally supports the needs of the users and the context, unconstrained by the current technological possibilities. The scenario thus brings together the continuous care concepts that all partners envision, the implications of the scenario for all project partners, and the process that was followed to attain the current iteration of the scenario. The sunny-day scenario consists of a number of scenes in which the actions of the personas are described in such a manner that the functionalities of the continuous care concepts become clear. In the project, the innovation binder serves as a boundary object [33], an instrument to unite both technical and user researchers and to create a common focus for all partners involved in the project. Since the evolution of the scenario, together with clear links to the insights obtained through the co-creation process and resulting implications, is also included, the innovation binder is closely connected to all three objectives of the ACCIO project.

2.1.4 Objective & Paper Organization

The goal of this paper is twofold. On one hand, the process we have designed to realize a participatory ontology engineering methodology in the ACCIO project is described. A series of workshops is described that were organized to actively involve domain experts in the ontology engineering process. The instruments that were used to unite all the insights gained during these workshops and how these were translated into the ontology are presented. On the other hand, the continuous care ontology, which resulted from applying the methodology, is presented.

The following specific research questions are addressed:

- **RQ1:** How to involve users in the creation of an ontology?
- **RQ2:** How to reach a cross-institutional validity of an ontology?

This paper elaborates on previous publications about this research [34, 35] by concentrating on the whole ontology development process by including later stages of the ontology construction and the workshops that have addressed the cross-institutional validity of the ontology. As such, a reflection on the whole process was possible.

The remainder of this paper is organized as follows. Section 2.2 describes the various user-driven methodologies, techniques and workshops used and evaluated for building a continuous care ontology. In Section 2.3, the developed continuous care ontology is presented. The developed co-design methodology is discussed as a whole in Section 2.4. Finally, Section 2.5 describes the lessons learned during the development of the methodology and highlights the most important conclusions and future work.

2.2 Ontology co-creation methodology

In this part of the paper, the steps are described that were taken in order to go through all the stages that Pinto and Martins [14] have identified for ontology development. These steps of the proposed participatory ontology engineering methodology are summarized in Figure 2.1. First, the observations that were made in two institutionalized care settings and the formation of a stakeholder group are introduced. Next, five workshop types are described in which domain experts, both targeted end-users and other stakeholders, were involved in various ways in the development of the ontology. The research team mentioned in these phases consists of all project researchers, both ontology engineers and social scientists. The descriptions of the observations and workshops are split up into the following parts:

- *Objective:* explaining the intended goal
- *Participants:* detailing the profiles of the people participating
- *Method:* conveying the general methodologies that were followed
- *Practicalities or Procedure:* diving deeper into the practical details of the organization of the workshop or observations
- *Analysis:* explaining how the output is structured, processed and analyzed
- *Results:* presenting the results obtained
- *Reflections:* reflects on the followed methodology and obtained results in respect to the objective

2.2.1 Composing a stakeholder group

To achieve a well-grounded, user-centered design process, a representative stakeholder group was composed at the start of the project. This way we wanted to ensure all relevant perspectives would be included in the process of the ontology creation as well as to create a stage where ontology engineers, who are concerned

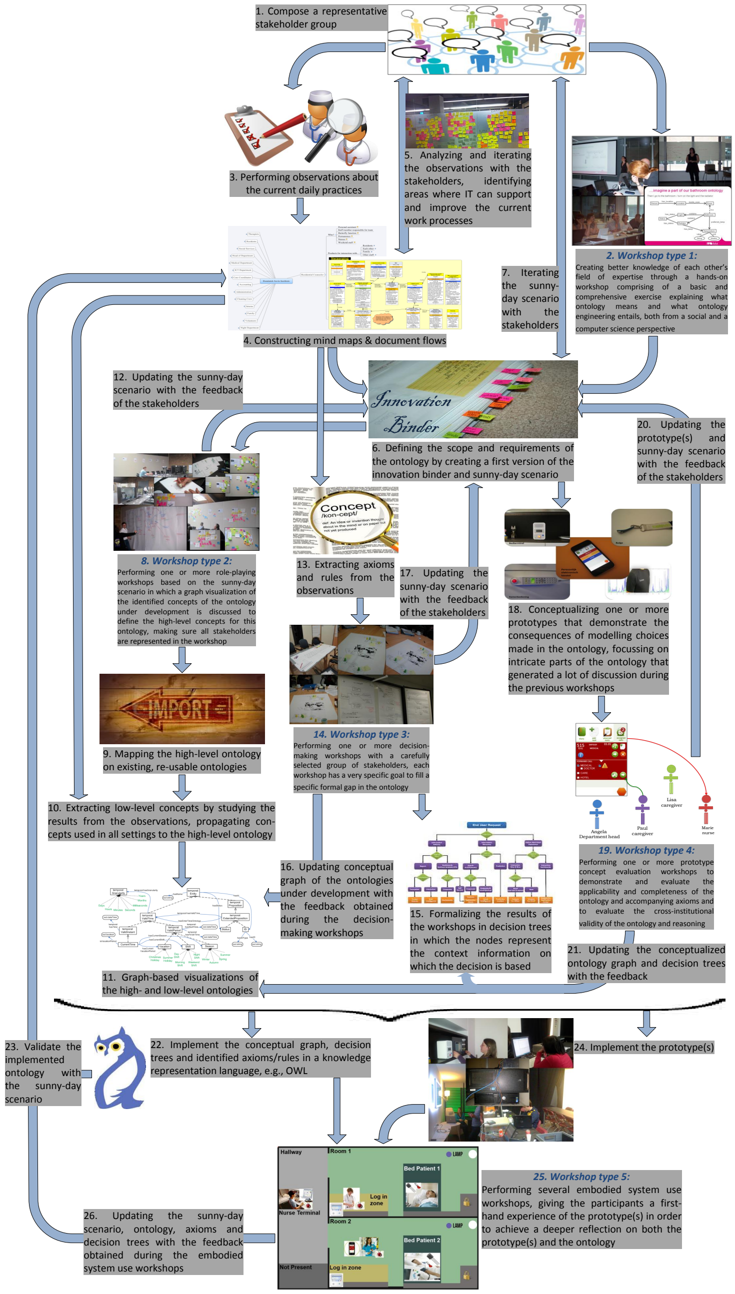


Figure 2.1: The various steps of the proposed participatory ontology engineering methodology

with the methods for building the ontologies, can interact with domain experts and social scientists. The stakeholder group consists of the following subgroups:

- Ontology engineers with a computer science background who have been involved in eHealth projects.

- Targeted end-users, i.e., staff from the observed settings and related settings, such as nurses, doctors and residential caregivers.

- Professionals working for the healthcare industry, e.g., representatives of a furnishing company, namely Boone International [36], and a company that makes communication systems, namely Televic Healthcare [37].

- Social scientists experienced with conducting user research.

Not only did the stakeholder group consist of people with different qualifications, but there was also a big difference in the extent to which the stakeholders 'believed' in the project objectives. Some took a very skeptical position, which is important to promote critical thinking and to ensure that the daily care practices are taken into account. However, others felt strongly positive about the objectives and took on the role of advocates of the project within the organizations they were affiliated with, thus creating goodwill within the participating care residence and hospital. These enthusiasts also dared to dream and think very future-oriented. We felt that finding a right balance between skeptical and enthusiastic stakeholders is important. A workshop with too much skeptics tends to become too focussed on current care and work practices. While the presence of too much enthusiasts causes the proposed solutions to be too far from daily practices to be adopted are not tuned towards the big problems the caregivers are daily faced with. While it is difficult to actively pursue a good balance in the stakeholder group, project members should be aware that having mainly skeptics or enthusiasts could have a considerable impact on the results. Therefore, it could be interesting to let the possible participants fill out a questionnaire, which reveals their view on technology and innovation. The results of this questionnaire could be used to compose balanced stakeholder groups.

2.2.2 Observations at institutionalized settings of care

2.2.2.1 Objective

The objective is to get a first understanding of the context the ontology would have to describe and how the provision of continuous care can be optimized.

2.2.2.2 Participants

Two major types of institutionalized continuous care settings were identified, namely residential (focus on caring for residents) and hospital care (focus on curing patients). A representative Flemish institution from each of these settings closely collaborated in the project, namely Dominiek Savio Institute [38], which provides residential care to people suffering from cognitive and physical impairments, and OLV Hospital Aalst [39].

2.2.2.3 Method

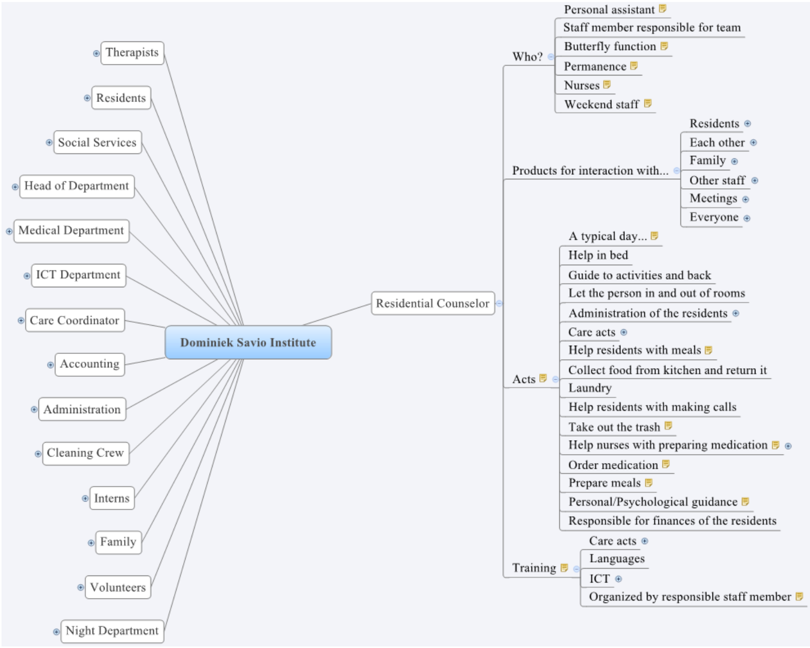
The researchers did a contextual inquiry [40] in the two settings. They identified the main roles in the organization and described their daily practices. Additionally, all documents used in the institution were listed and their flow in the institution was mapped. The observations were performed by the social scientists together with ontology engineers.

2.2.2.4 Practicalities

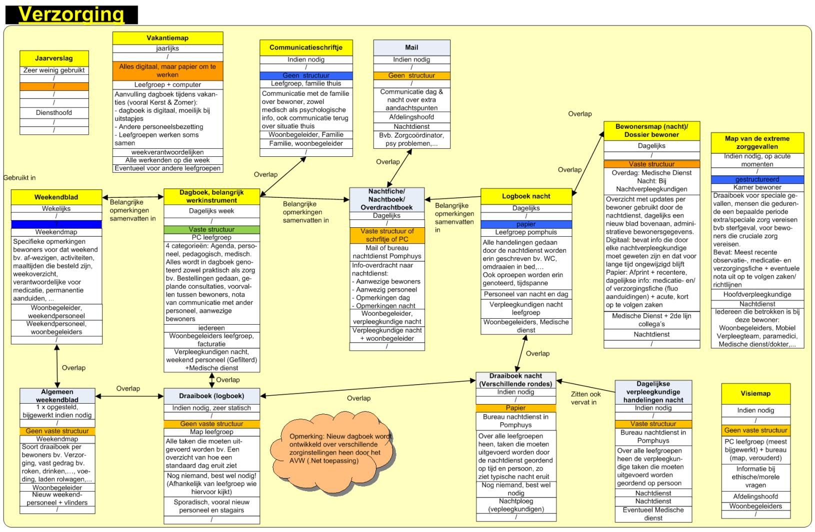
Daily care practices, e.g., helping a person go to bed or communicating the status of the care receiver to other staff members, were observed in both settings. During two weeks, 12 day shifts and 2 evening shifts were observed in the care residence and in the hospital 6 shifts were observed during three days. During every shift, two researchers were present. All roles, e.g., medical staff, care receivers, therapists and coordinators, per setting were identified and representatives of each role were followed and interviewed. Extensive notes and recordings were made during the visits.

2.2.2.5 Analysis

All insights were synthesized in extensive mind maps [41]. A summarizing example is shown in Figure 2.2a. To get a clearer view of all the information being exchanged during the continuous care practices, separate files were made to structure all documents and their flow. This document flow outlines all documents (analog or digital) and formats of communication used during the daily continuous care. It also describes the properties of the documents, such as the content, the specific aim, the author(s), the target audience, the storage location and the relationships to other documents. To get an idea of the structure and purpose of this document flow, a small part is visualized in Figure 2.2b (in Dutch). Both the mind maps and document flow were evaluated and validated with different stakeholders from the institutions, who were not necessarily involved in the observations.



(a) Example mind map



(b) Part of the document flow, visualizing the information being exchanged

Figure 2.2: Mind map and document flow, capturing the results of the observations at Dominiek Savio Institute

2.2.2.6 Results

Representatives from the stakeholder group, i.e., ontology engineers, social scientists, end-users and R&D engineers working for the healthcare industry, processed, discussed and evaluated the constructed mind maps and document flows to identify areas where IT could ideally support care receivers in their daily activities and caregivers in the provision of continuous care. Problem areas and possible solutions were captured on post-its during brainstorm exercises. Next, each participant received a number of votes to put towards the problems & solutions, which he or she perceived as most important or desirable. The post-its were then ranked according to the number of votes. From the top ranked ideas, the ones were selected that were feasible within the time-frame of the project, alligned most with the aim of the project and expectations of the project partners.

As a result, two areas for improvement were selected: namely *information integration & data provisioning at the point of care* and a *dynamic nurse call or care request system*. The first area can be summarized as providing the right information at the right time, at the right place for the right person. This entails an increased need for *mobile* services to support data input, e.g., care registration, and requesting data, e.g., medical data about care receivers, at the point of care. Moreover, information should be *integrated*, *prioritized* and *filtered* based on *contextual* information. Examples of this include only monitoring values that violate a threshold or only displaying recent information that pertains to the care receiver a staff member is currently tending to.

To demonstrate the possibilities of an ontology-based, context-aware application, a more dynamic nurse call system was chosen to be developed as one of our future continuous care concepts. This system should use the integrated information about the staff and care receivers to find the most appropriate staff member to handle a request of care and at the same time optimize the current workflow in a *dynamic* way. The current nurse call systems are mainly lacking the ability to adapt, to give *feedback* to the person requesting care and to give an *overview* of the current situation, e.g., localization of care receivers and staff. By adressing these shortcomings, this future care concept tackles both areas for improvement as previously identified.

The insights in the mind maps were combined with this research objective to write a first version of the sunny-day scenario and the innovation binder. The most important created persona was Erik. Erik lives at a care residence, has Duchenne disease and is dependent on a wheelchair. His main wish is to increase his autonomy. Personas were also created for Erik's parents and brother, staff at the care residence and associated hospital. The constructed sunny-day scenario describes the story of Erik who becomes unwell in his residential care setting and is transported to a hospital for treatment. Both the care for the resident in the residential care setting and the hospital are supported by the dynamic nurse call or care request system, which uses the ontology to gather information and infer knowledge. As such, the mind maps and document flow served as input to define the scope and purpose of the ontology.

2.2.2.7 Reflections

Having the observations in the care residence and the hospital being conducted by social scientists and ontology engineers ensured a 'naive' perspective on the practice of giving care. While domain experts have a much deeper understanding of the domain of care than observers can achieve after a few days, or even weeks, they do also tend to overlook the obvious. While experts are obviously required to gain a good understanding of a domain, it is also useful to include observations by laymen.

The observations and ensuing discussions allowed us to construct draft versions of the innovation binder, define the scope and purpose of the ontology and determine the focus on dynamic nurse call or care request systems. As such, the observations were the perfect starting point for the series of workshops that would aim to elaborate on them by including stakeholders and targeted users in various ways.

2.2.3 Workshop type 1: Introducing ontologies

2.2.3.1 Objective

The goal of this workshop is to educate the stakeholders about ontologies and ontology engineering, to disucss how to jointly design ontologies in the domain of institutionalized care and to create a shared perspective on ontologies.

2.2.3.2 Participants

The participants consisted of members of the stakeholder group, in total 22 persons, including some targeted end-users from the observed settings such as a caregiver and a doctor.

2.2.3.3 Method

After being informed of what an ontology is, the stakeholders were asked to do some small ontology construction exercises. The group was then stimulated to think about how ontologies could support care processes.

2.2.3.4 Procedure

The half-day workshop was organized by both ontology engineers and social science researchers. It started with a description of what an ontology is, both from a social science perspective as from an ontology engineering perspective. In order to increase the participants' understanding of ontologies, they were asked to do a small exercise as preparation for the workshop, namely to make a textual description of the bedroom as a concept. We presumed that the bedroom would be an important space where we want the ontology to be active in a residential and hospital healthcare environment. During the workshop, they had to mark in this textual preparation which sentences and words were important to define the bedroom. This made clear that different perceptions exist of what a bedroom is and that a common agreement is needed in order to create a model or ontology of a bedroom. The ontology engineer then restructured terms into concepts in a graph defining subtypes while explaining essential terminology of ontology engineering. To explain the terminology an example ontology of a bathroom was used, as shown in Figure 2.3. It was shown how this graph could form the basis for an ontology, and how defining constraints and relations can limit the different interpretation of the ontology concepts. Finally, the ontology engineer illustrated how an application can potentially be built using this ontology. Once the participants had a better understanding of what ontologies are, a group discussion was started on how these ontologies can be used to automate processes in institutionalized care. Some impressions from the workshop are visualized in Figure 2.3.

Figure 2.3: Impressions of workshop 1: Introducing ontologies

2.2.3.5 Analysis & Results:

This workshop had no tangible results, since it was mainly organized to inform all stakeholders on ontologies and to create a shared understanding of what ontologies are and what their role should be in this project. The participants did indicate their understanding of the project in general and of ontologies in particular had thoroughly improved, to the extent that they would even be able to explain the concept of ontologies to others.

2.2.3.6 Reflections

The participants seemed generally enthusiastic about the hands-on exercise and discussion that ensued. The most important consequence of the workshop surely was that common ground had been created for all project partners. This is an essential step in interdisciplinary research and co-creation processes as the group diversity and the background knowledge is very different. A first essential step is learning to understand each other, so a better integration of concepts and views can be initiated in the subsequent development steps. After this workshop, the concept of ontologies was no longer abstract and vague. It was now clear and all stakeholders shared a common purpose in the project. This in turn would enable the stakeholders to step up as project advocates, and motivate other stakeholders to take part in the subsequent workshops.

The researchers had considered repeating this workshop with the targeted end-users only, i.e., people from Dominiek Savio Institute and OLV Hospital Aalst. We thought this might give them additional insights into why the workshops would be conducted and which results we wanted to obtain. However, we refrained from doing so because this might be too demanding on their part and would resemble the HCOME approach, something we wanted to avoid. Moreover, we wanted to evaluate how we could construct an ontology with domain experts, without them actually needing to know what ‘ontology’ means.

2.2.4 Workshop type 2: Scenario role-play

2.2.4.1 Objective

The two observed continuous care settings, namely Dominiek Savio Institute and OLV Hospital Aalst, are rather different, but the aim is to construct an ontology that captures the continuous care practices in both settings. Moreover, the ontology should not only be applicable to the two specific settings studied during the observations. On one hand, the ontology should be abstract enough to be applied to all continuous care settings, while on the other hand it should remain specific enough to build practical applications on it. Therefore, it is important to identify the *high-level concepts* that are used with the same meaning within the continuous care domain. The goal of this workshop was thus to construct the basis for a high-level ontology that describes the concepts and the relations between these concepts in institutionalized care settings.

2.2.4.2 Participants

Since our goal was to capture the concepts that had a shared meaning across different care settings, we specifically invited participants from care residences as well as hospitals. The 18 participants were recruited from the observed settings, but also from other hospitals and care residences, representing different subgroups ranging from caregiving to managerial functions.

2.2.4.3 Method & Procedure

The workshop was organized in the Patient Room of the Future (PRoF) [42], which is a high-fidelity mock-up of a near-future patient room integrating innovations from soft- and hardware developers as well as furniture. The PRoF consists of a typical patient room and hallway found in a hospital setting, as well as a room mimicking a homecare setting. The workshop proceeded in four steps:

1. ***Storyboard:*** The workshop began with a short introduction to the ACCIO project and its goals in layman's terms, e.g., avoiding the term 'ontology'. The first version of the scenario in our innovation binder was visualized in a storyboard. This storyboard was hung on the wall, as can be seen in the top left corner of Figure 2.4, leaving plenty of space in-between the different scenes of the story so additional comments or enhancements could be added. The storyboard formed the common thread in the workshop and served as a point of reference during group discussions.

2. ***Role-play:*** The participants took turns in playing out each separate scene from the storyboard within the actual patient room. Persona, e.g., an impatient care receiver or a non-medical caregiver, and situation, e.g., busy night or visiting hour, cards were used to inform the performing participants about the context and character they would be playing. The remaining participants acted as an audience, observing a live feed of the role-play on monitors set up outside the patient room and noting down issues or remarks.

3. ***Reflection:*** After each scene, the performers and audience were reunited to have a group discussion on the events that occurred during the performance. The moderator inquired into the remarks the audience noted down and added these to the storyboard by means of post-its or new drawings made on the spot by one of the researchers with graphic skills, pointing towards perceived problems, e.g., difficult for care receiver to alert a caregiver, or opportunities, e.g., automatic fall detection. Examples of these post-its and drawings are visualized on the right of Figure 2.4.

4. ***Modeling:*** During the performances and group discussions, two ontology engineers followed the entire process in the background, translating its out-

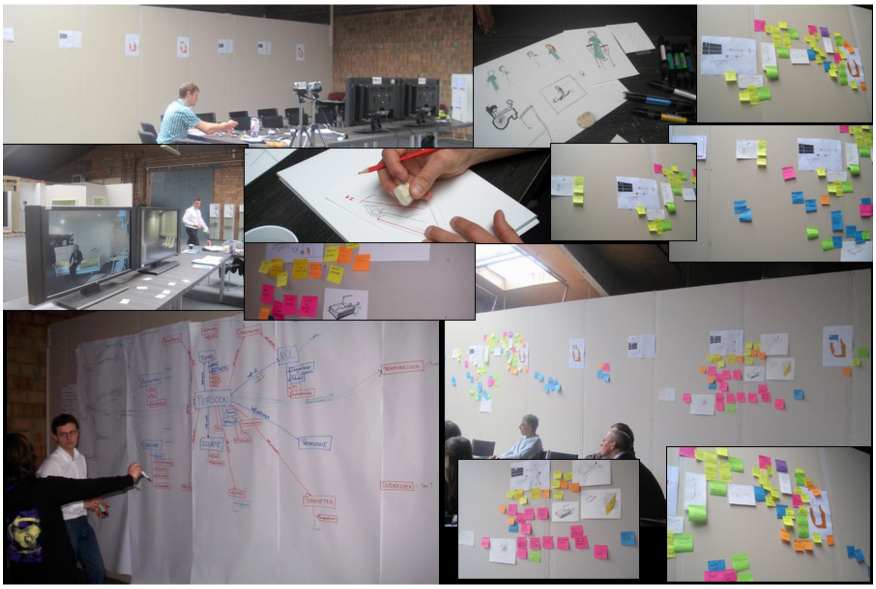


Figure 2.4: Impressions of workshop 2: role-playing at the PRoF

put in order to construct a high-level ontology in the form of a graph. This is shown in the bottom left corner of Figure 2.4. The ontology concepts (nodes in the graph) and relations (vertices of the graph) were drawn on clearly visible posters next to the storyboard so that all stakeholders could see its evolution and provide feedback if necessary.

2.2.4.4 Analysis

After the workshop, the researchers collected all the notes and drawings attached to the storyboard, clustering the material belonging to the respective scenes. These clusters were further analyzed in a follow-up meeting, which led to the identification of general themes, e.g., distinct attributes of different caregivers, the importance of roles and competences and desired sensors and related data extraction, which in turn were integrated in the ontology.

2.2.4.5 Results

The workshop resulted in a new iteration of the innovation binder scenario, taking into account the feedback of the stakeholders. Additionally a first version of the high-level ontology was constructed, which is illustrated in Figure 2.5. Six major sub-domains were identified, namely information pertaining to tasks & processes, to person profiles, roles & competences, to medical context, to sensor observations, to temporal information and to general concepts. The most important concepts within these sub-domains were identified (69 concepts in total). This allowed the ontology engineers to integrate most of the information obtained during the observations in this ontology, based on the mind maps and document flow. Based on this ontology, an investigation of existing ontologies was performed to evaluate if these could be re-used. The existing ontologies, which were found suitable, were imported. These included the Wireless Sensor Network (WSN) ontology [43], the OWL-S Process ontology [44] and the SWRLTemporalOntology [45].

As mentioned previously, the two settings under scrutiny during the observations are quite different. The major difference between the two is that residential care settings focus more on care, while hospitals focus more on curing the patients. Therefore, it was decided to conceptualize two low-level ontologies, containing concepts specific to these settings. These low-level concepts are always subconcepts of concepts in the high-level ontology.

The low-level ontologies were constructed by analyzing the document flows and mind maps and extracting concepts such as roles, competences, tasks (care acts) and profiles. The two resulting low-level ontologies were compared to identify common concepts used within both settings with the same meaning. These concepts were moved to the high-level ontology. The resulting ontologies are discussed in more depth in Section 2.3.

2.2.4.6 Reflections

Although this workshop did help us to validate and extend the proposed innovation binder scenario, the process of constructing a high-level ontology was not fully satisfactory. Discussions covered too many topics and a consensus was rarely reached. This made it hard to evaluate if a certain concept was used and whether it had the same meaning for the different stakeholders. It was also difficult to formalize or pick up the different relations that existed between concepts that arose.

After the workshop, the participants were asked to fill in a formal, anonymous evaluation form. The mixture of profiles ensuered that the participants evaluated the workshop as very interactive and found it to be an enriching experience. On the other hand, the session was deemed a little too time consuming.

The researchers echoed the latter remark. The obtained results were useful, however not in relation to the invested amount of (preparation) time. In hindsight, the broad and time-consuming discussions were also a consequence of the rather large number of participants. As not all participants took part in workshop 1, they did not have as much insight in the project's objectives and the concept of ontology engineering. The lack of contextual information combined with the large size of the group probably explains the lack of collaboration between the participants and ontology engineers. The ontology engineers rather observed the activities than

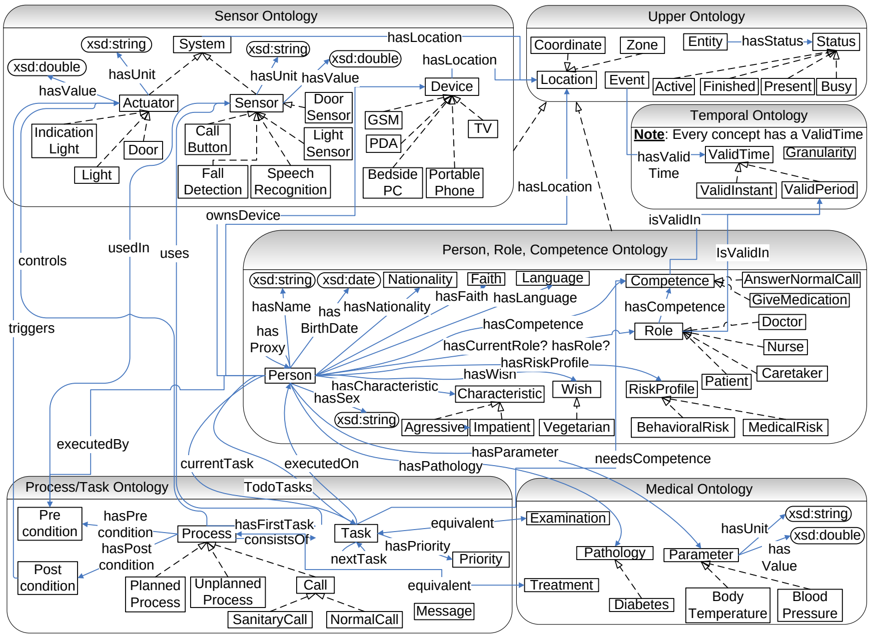


Figure 2.5: The high-level ontology resulting from workshop 2: scenario role-play

collaborated with the participants to co-create the ontology.

The lack of consensus was however not problematic in se, although it did seem so at first. In retrospect, it became clear that a lack of consensus rather is an indication that the concept where no agreement can be found, should not be part of the high-level ontology.

In sum, it seems that this type of workshop is more suitable for validation than conceptualization.

2.2.5 Workshop type 3: Decision-making

2.2.5.1 Objective

As explained in the Results subsection of workshop 2, a new version of the high-level ontology and first versions of the low-level ontologies were obtained by integrating the results of workshop 2 with the observations. At this point, the ontology was still a conceptual model. The next step involved translating this conceptual model into a formal one. This means that the concepts are defined through *axioms* that restrict the possible interpretations for the meaning of those concepts, e.g., defining which competences a certain role has, which competences are needed to perform a task or which sensor observations are valid. A lot of these restrictions were derived from the data extracted during the observations or the iterative disccussions to construct the innovation binder and sunny-day scenario. However, there were still some formal gaps in the ontology. To capture these axioms and restrictions, workshops of type 3 were organized to capture these decision processes. Ontology engineers could then translate these processes into additional ontology concepts and rules/axioms. For the continuous care ontology under development it was unclear how the different possible care requests were restricted in the priorities and reasons they could get and to which caregivers they should be assigned.

2.2.5.2 Participants

To make sure that the context-dependent factors are captured, a separate workshop was held for the two types of institutional care settings, i.e., care residences and hospitals. The workshops took place in the observed settings. The workshop in the care residence welcomed 16 participants, including caregivers as well as stakeholders holding a coordinating or managerial function. The workshop in the hospital setting was scaled down to 4 members. A nurse, head nurse, caregiver and doctor participated to ensure a mixture of profiles.

2.2.5.3 Method & procedure

The decision-making workshops were organized on-site in the observed care residence and hospital and took about 2 hours to complete. Other than the number of participants, the procedure was alike in both instances. The workshop began with asking the participants to describe on paper a complex situation pertaining to the formal gap in the ontology we were adressing, in this case assigning and prioritizing care requests or nurse calls. After this individual exercise that took about 5 to 10 minutes, a blueprint of the physical working environment of the participants was introduced. The complex situations that were noted down by the participants served as input for the rest of the workshop.

Participants were asked to suppose that they were an intelligent system that had a complete overview of the institution and current situation and context. This system is responsible for solving the task pertaining to the formal gap in the ontology, i.e., in this case finding the most appropriate staff member to handle a care request. Other tasks can easily be devised for filling other formal gaps, e.g., assigning tasks to staff members to determine the mapping of competences on roles and the mapping of competences on tasks. The real life situations described by the participants were selected to further discuss how this intelligent system should ideally handle them, i.e., prioritize and assign nurse calls in a broad sense. Each situation started with a very limited initial setup that was illustrated on the blueprint of the institution. Different props were used to represent staff members, care receivers and material, as can be seen in Figure 2.6.

To make a sensible decision, the participants, playing the role of the system, could collect additional information about the situation by asking questions, e.g., how many staff members are present in the department? What are their roles? Who made the request? Instead of immediately giving an answer, a discussion was first held about the importance of the requested information. We did this by consistently asking three questions: (1) Why do you feel the answer to this question is pertinent? (2) Does everyone agree? (3) Can you give examples of answers to this question?

By posing these questions, prior to giving an answer, the researchers and ontology engineers could tap into the reasoning made by the system, i.e., the participants, and minimize their own influence on the decision making process. Finally, the question was answered and visualized by adding or displacing the relevant props on the layout, after which a next question would further unravel the situation.

The questions and the order in which they were asked, gave the ontology engineers insights into the needed information and the importance of this information for making a decision. The ontology engineer processed the outcome on paper in the form of a decision tree, as shown in the bottom right corner of Figure 2.6. The order of the information in the tree reflects its importance, while the different nodes represent high-level concepts to which the questions of the participants pertained, i.e., location or task priority, as well as the relations or interdependencies that existed between these concepts.

2.2.5.4 Analysis & Results

After the workshop, the preliminary decision trees were transformed to more formal decision trees for each setting. After both workshops, the decision trees were compared to extract common parts. The common parts delivered input for the high-level ontology, while the other branches are encoded in the low-level ontologies. Finally, the decision trees were translated to concepts, axioms and rules in the implementation stage of ontology creation.

As an example, Figure 2.7 shows a part of the common decision tree that finds the most appropriate staff members to handle a call or care request based on the data in the ontology. Words indicated in bold refer to concepts from the high-level ontology. The algorithm first determines whether the call is an urgency call. Each care setting has its own procedure for handling these highest priority calls. These procedures are encoded in separate, context-dependent decision trees to which this decision tree refers. Otherwise, it is determined whether the call is a normal, i.e, made by a care receiver, assistance, i.e., a request for help by another staff member, or context call, i.e., launched by a sensor, which detected an anomoly. Next, the decision tree determines whether the reason for the call is medical, care or a 'hotel' task, e.g., a request for a glass of water. Based on this classification, staff members are sought with the required competences for carrying out such tasks. If there are

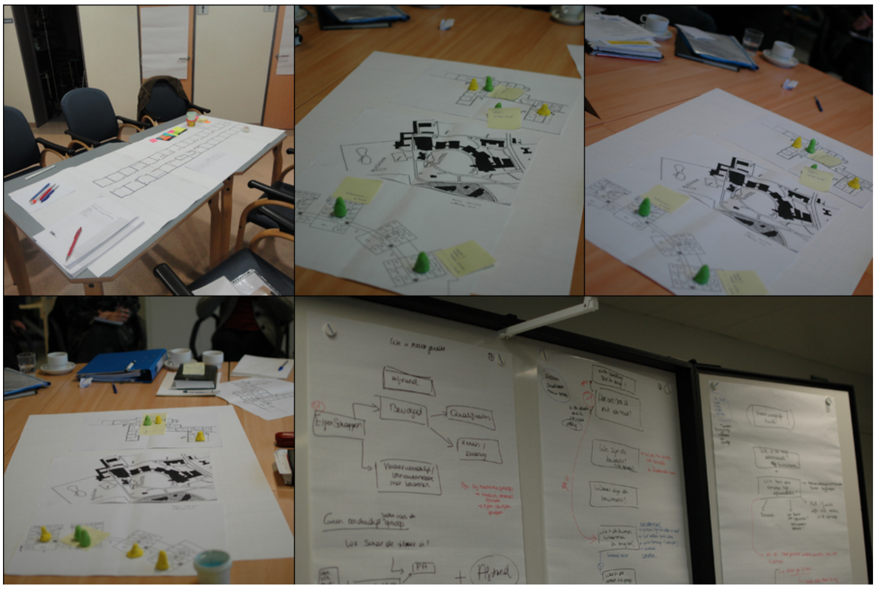


Figure 2.6: Impressions of workshop 3: scenario is visualized on a layout and high-level concepts, derived from the requested information, are summarized in a decision tree

multiple staff members available with the required competences, the decision tree assigns the most appropriate staff members to the call based on their current task, location, trust relationship with the care receiver and the priority of the call.

The workshops resulted in a new iteration of the innovation binder scenario, taking into account the feedback of the stakeholders.

2.2.5.5 Reflections

After conducting the first decision-making workshop in the residential care setting, it became clear that the participant group should be kept smaller. Besides the fact that some participants tended to dominate the discussion, we noticed that the large group and presence of managerial functions prevented the actual work processes being uncovered. Participants were inclined to give desired answers that corresponded with institutional or legal procedures, even if, for different reasons, these procedures could not always be followed. However, for our call or request system and underlying ontology to be usable, it is imperative that it is mapped onto the actual work processes, even if they digress from official procedures. Additionally, the larger group made it more difficult for the participants to follow the proposed workshop format and act as one system that needs to resolve the visualized situation on the blueprint.

At the second workshop, this problem was further resolved by printing out a

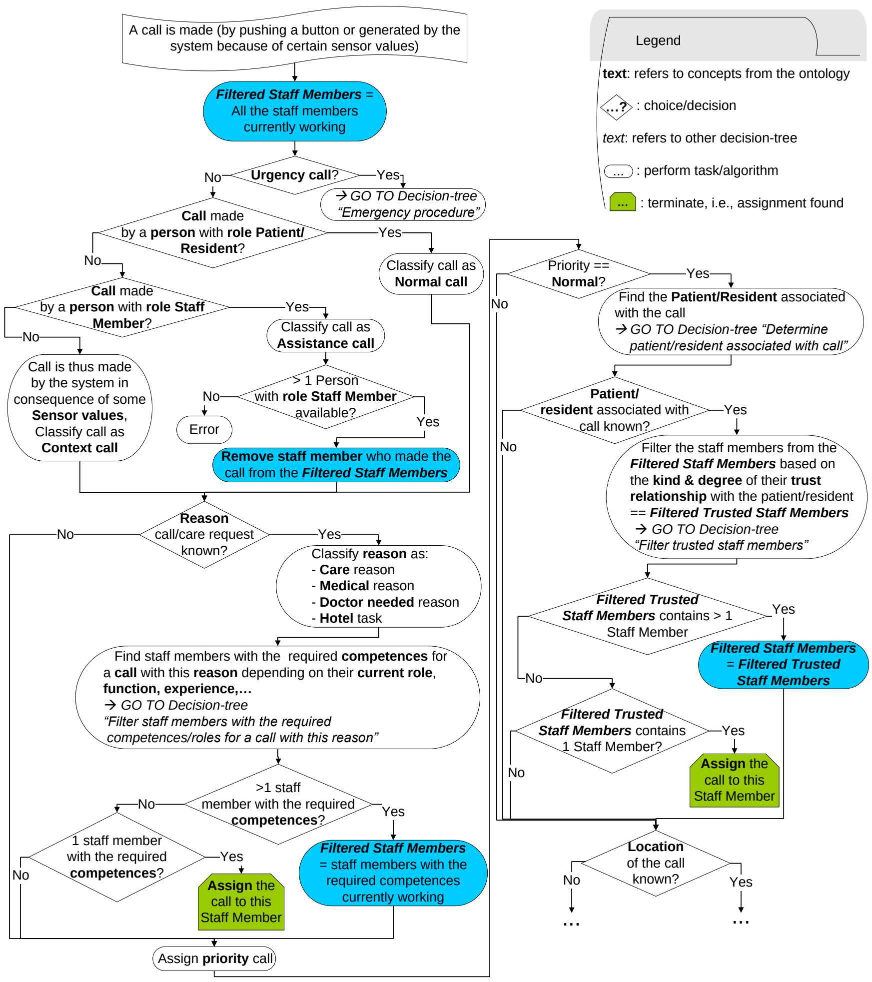


Figure 2.7: Decision tree to find the appropriate staff member to handle a call or care request

much larger blueprint so that everyone around the table felt essentially as equal players during a board game.

Ultimately, we could derive a lot of concepts on which decisions were based. For most of these, consensus was reached as to whether they were important to take into account. Prioritization turned out more difficult, as this was situation dependent and it was hard to derive what about the situation influenced it.

2.2.6 Workshop type 4: Concept evaluation

2.2.6.1 Objective

Based on the decision trees and conceptual ontology model achieved during the previous workshop, the prototype of a dynamic nurse call system was conceptualized. As mentioned in Section 2.1.3, this prototype was being built to demonstrate and evaluate the applicability and completeness of the ontology and accompanying axioms. The purpose of this workshop type was a concept evaluation of the nurse call system prototype under development, and to ensure cross-institutional validity of the ontology and reasoning.

2.2.6.2 Participants

In order to facilitate cross-institutional reflection, the recruited participants had various backgrounds and came from all types of institutions, both cure and care. A couple of staff members from the observed settings also participated. There was considerable variation in the professional qualifications of the participants, whose professions ranged from nurses or caregivers to doctors, designers and domain experts. In total, 14 persons participated in the workshop. Additionally 4 researchers were involved in the practical organization of the day.

2.2.6.3 Method

The conceptual prototype of a dynamic nurse call or care request system was presented to the participants in an instructional movie. For the discussion of the specific functionalities of the dynamic system, the group was split into two smaller groups for which two moderators guided the conversation.

2.2.6.4 Procedure

The participants were presented a 'slice' of the full scenario that described the dynamic nurse call system. The selection included several new functionalities that the research team had introduced, based on the insights gathered during the first three workshops. A movie was made in which these functionalities of the nurse call system were depicted. Lay actors played out the scenario in the high-fidelity mock-up of PRoF. The researchers chose to make a 'silent' movie where the depicted action was explained by intertitles, in order to avoid that the lay actors would struggle with the dialogue. It took half a day to film. The editing took about a full day. Screenshots of the personal electronic device the caregivers make use of in the scenario were also shown in the movie. Some minor improvements to the script were made on the spot.

In the 2,5 hour workshop, the movie was first played entirely, and then repeated and paused every time a new functionality got introduced. This functionality became the specific subject of a group discussion. For each discussion, the group was split up into two smaller groups, guided by two moderators. All remarks and insights were noted down on big posters. When the two groups reunited, the moderators presented the content on the poster to the other group. Additional remarks or insights that ensued from the discussion were added to the posters. The whole workshop was filmed.

2.2.6.5 Analysis

Afterwards, the involved researchers had a small discussion, listing the issues that they had observed during the workshop. Later, all researchers of the team looked at the video data separately, and wrote down their observations and thoughts. These were then brought together and discussed in a meeting with the others.

2.2.6.6 Results

The workshop resulted in some nuances with regard to the algorithm of the nurse call system. For instance, the idea of triaging calls generated a lot of enthusiasm, but raised concerns with regard to reliability. It also became clear that views on trust relations differ between institutions and have a dissimilar impact on work practices. These insights did not require an update of the ontology the nurse call system was based on, but the decision trees were adapted and the accompanying rules and axioms in the ontology were updated. Adjustments to the prototype were also made, as the participants also had some comments on the user interface of the application on the personal electronic device.

2.2.6.7 Reflections

The general impression was that visualizing the scenario as a movie stimulated the discussion considerably. The researchers had however somewhat underestimated the novelty of the nurse call or care request system. Especially since the participants had been widely recruited for cross-institutional validation and varied strongly in their professional profile. Even though some background on the project was given at the start of the workshop, some participants had difficulty grasping the full range of the project. More information about the project, the field research, the preceding workshops, and insights the new system was based on, should have been given at the start of the workshop. Now, while the movie introduced all functionalities gradually, it still felt like too much new information was given at once.

The workshop resulted in considerable feedback for the algorithm of the nurse call system and thus the axioms & rules captured in the ontology. However, no changes were made to the concepts and relations of the ontology after this workshop. This could mean that the method was less suited for ontology feedback, but it can rather be understood as a confirmation of the ontology. Although the workshop was certainly focused on the nurse call system, missing concepts in the ontology or false relations between concepts in the ontology should have come up during the discussions. Therefore we are inclined to see the lack of input for the ontology as a confirmation of the ontology, rather than concluding the method was unsuited for cross-institutional ontology evaluation. To make sure participants thoroughly reflected on the system, and the concepts and relations that were used in it, an additional workshop type was organized in which the participants are given a profounder experience of the ideas included in the ontology and the prototype.

2.2.7 Workshop type 5: Embodied system use

2.2.7.1 Objective

After the previous workshop, the conceptual graphs and decision trees were translated into an ontology language. As the goal was not to overburden the stakeholders, they were not actively involved in the implementation of the ontology. However, to prevent losing sight of the user requirements when making implementation decisions, the ontology engineers constantly validated the ontology under development with the scenario in the innovation binder.

The Protégé editor [46] was used to develop the continuous care ontology in OWL. The Pellet Reasoner [47] was used to check the consistency and the classification of the ontology. The Semantic Web Rule Language (SWRL) [48] was used to express rules using concepts from the ontology. The resulting continuous care ontology is discussed in Section 2.3.

Based on this ontology, the prototype of the dynamic nurse call or care request system was implemented as described in [31].

The workshop of type 5 aimed to give the participants a first-hand experience of the prototype in order to generate deeper reflection on both the application and the ontology on which the application was built.

2.2.7.2 Participants

All participants worked in a care residence or a hospital (not necessarily the observed ones), or did so until recently, for instance, teachers at a nursing school. They all had first-hand experiences with some sort of nurse call or care request system. In total, 15 participants were involved in these workshops.

2.2.7.3 Method

After an elaborate introduction, the participants were asked to try out some of the functionalities by role-play. The participants were given persona and context cards and asked to play out seven scenes. In between and afterwards, the participants and the researchers discussed the application.

2.2.7.4 Procedure

These embodied try outs required the test users to do a role-playing game in the PRoF in order to have a deep experience of the implemented prototype. The setup of the prototype in PRoF is visualized in Figure 2.8. For the prototype, RF tags and receivers were integrated to track the locations of the care receivers and staff. Temperature sensors were also available to monitor the temperature of the care receivers. The developed ambient-intelligent nurse call system was installed in PRoF and integrated with the available light control system, RF tags and sensors. Smartphones running the designed mobile application were also provided. This prototype allowed users to experience a fully immersed, more profound, contextual experience of the system in a lifelike context. As we wanted the participants to have a complete experience of the system, we invited small groups, i.e., two or three users per workshop, so that they would be occupied at all times and the researchers could follow them one-on-one. As such, seven workshops were organized for 15 participants. All workshops were entirely filmed.

During a 2.5 hour workshop, the participants were first given an elaborate introduction, with two movies framing the research project and informing the users of the functionalities of the new system. Two researchers then interviewed them on their experiences with nurse call or care request systems, and asked them about the limitations they experienced with the system they were familiar with. Next, the participants were given an introduction of the personal electronic device, learned how they could accept calls, forward calls and call the care receiver or a colleague. The participants were then asked to play out seven scenes in the PRoF. For each scene, a test user received a persona card and a context card, informing the test user of the role he or she would have to take up and what he or she would have to do. After scene two and five, a small pause was held to discuss the functionalities of the system. Afterwards, a more elaborate discussion was held. The functionalities that were introduced during the workshop included: location detection, occupation, trust relations, call forwarding and calling a care receiver or a colleague.

2.2.7.5 Analysis

During the workshops, the researchers immediately noted down the first emerging insights. These were gathered in a draft document. Since all workshops were

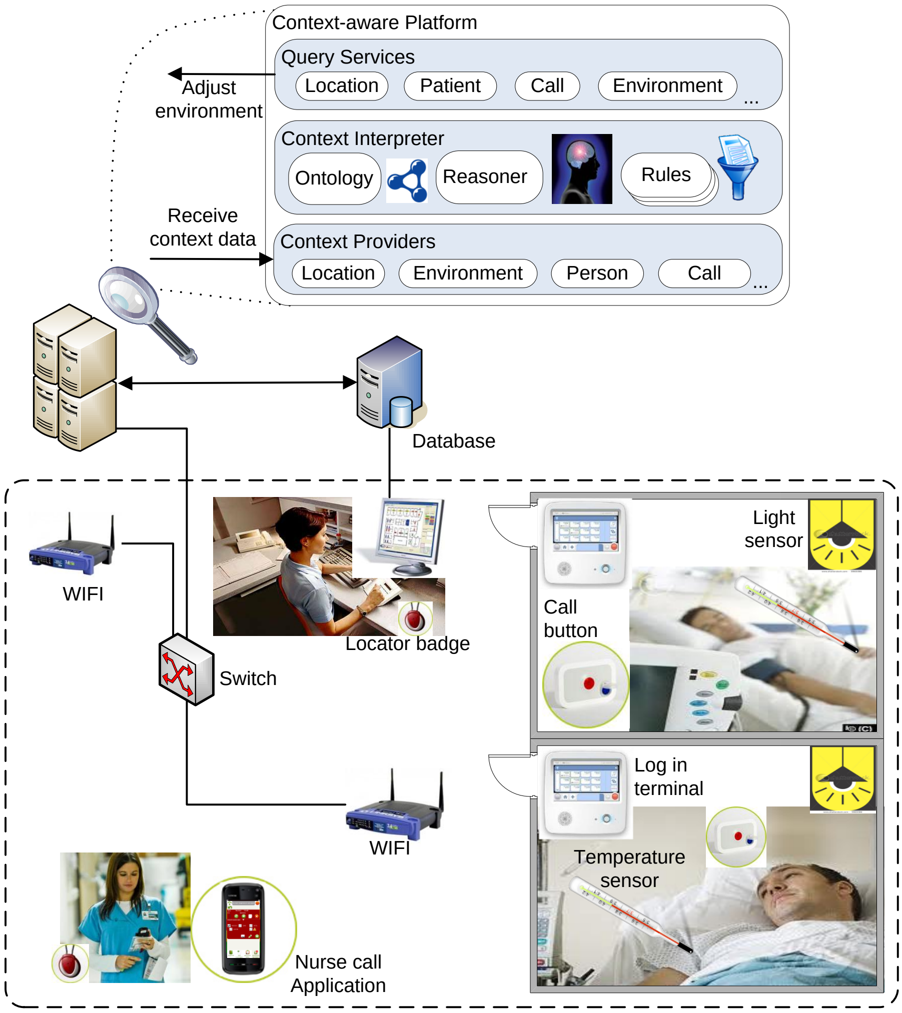


Figure 2.8: The setup of the nurse call prototype in PRoF for workshop 5: Embodied system use

filmed, these workshops resulted in a considerable amount of video data. Therefore, the video data was split among the researchers, who went through their part of the data on their own and wrote down their insights. These were then brought together and clustered.

2.2.7.6 Results

Just as in workshop 4, these embodied experiences and the ensuing discussions resulted in more feedback for the application than it did for the ontology. For instance, some missing functionalities of the nurse call or care request system became clear. Also, minor remarks with regard to the interface of the personal electronic device were made. Again, a lot of comments were made on the concept of the inclusion of the trust relation in the system. Additionally, some concerns were raised with regard to the impact the system would have on the care receivers' expectations towards the caregivers. Some important findings were also collected in relation with the triage system and the changes in the work practices that it would ensue. Finally, it gave the engineers insights into the pitfalls, difficulties and doubts that have to be taken into account when introducing this ontology-based application in the working environment.

2.2.7.7 Reflections

After workshop type 4, the researchers realized a more elaborate introduction to the ontology and its application was needed. Therefore, a considerable amount of time, at average about 30 minutes, of the workshop was reserved for introduction. This seemed to improve the participants' understanding of the system, but during the scenes, additional remarks were sometimes still needed. In fact, the role-play increased the participants' understanding of the system most, and it was only after playing a few scenes that some participants fully grasped the system's operation. Similarly, while most participants expressed to be impressed after a first glance at the system, a more thorough experience of the system made them more critical.

The researchers were pleasantly surprised by the easy uptake of the role-playing exercise by the participants. Even the rather reserved participants, got more engaged in the exercise after playing out a few scenes. Therefore, we feel that these workshops were a success, despite a high investment of time of all researchers involved. The workshop gave the researchers much feedback on the algorithm of the prototype and only minor feedback on the concepts/relations in the ontology. Again, this could be due to the fact that the ontology already contained most concepts and relations in the cure and care field, or due to the fact that the method was not suited for ontology validation.

This is not to say that the described user involvement could not be improved. Although the final tests took place in PRoF, which was very close to reality, it was felt that a real-life setting could generate further insights. It will be investigated how a mobile set-up of the system can easily be tested in a real-life work setting. However, the varying available technology and networks make this a challenging endeavour. During the final tests, some technical issues popped up, which threatened to reduce the user tests to technical tests. This was to be expected, as

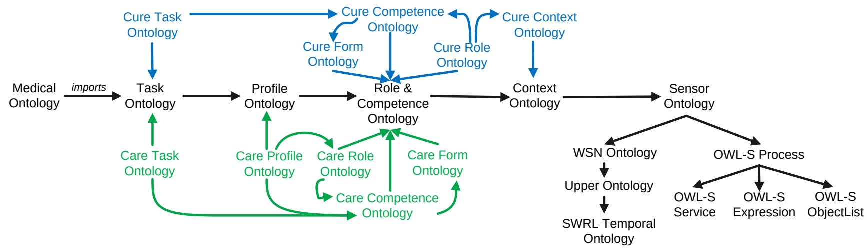


Figure 2.9: Import schema of the high-level continuous care ontology (black) and the low-level care (green) and cure (blue) ontology

the implemented application is merely a prototype. Although these issues were quickly solved, it was sometimes hard to distinguish the participants' feedback on the system from feedback related to technical system failure.

2.3 The co-created continuous care ontology

As mentioned previously, the main difference between residential care settings and hospitals is that the first focus more on care, while the latter focus more on curing the patients. Consequently, the ontology was split into a high-level ontology, modelling the concepts that are used with the same meaning within the continuous care domain, and two low-level ontologies, one focusing on care and the other on cure. These low-level ontologies contain concepts that are used with a different meaning in both settings or that are used within residential care, but never within hospitals and vice versa. The following sections give an overview of the high-level and low-level ontologies.

2.3.1 The high-level continuous care ontology

As mentioned in Section 2.2.4, 6 sub-domains of the high-level ontology were defined during the role-play workshop. These were largely maintained during further conceptualization of the ontology. However, the sub-domain pertaining to person profiles, roles and competences was split up in 2 sub-domains: one concerned with people and their characteristics and one focused on roles and competences. Moreover, the general sub-domain was split up into the context ontology and the upper ontology. The upper ontology also subsumes the temporal sub-domain. The import schema of all the ontologies that make up the high-level ontology is visualized in Figure 2.9. The seven parts of this ontology are presented in detail in the following subsections.

2.3.1.1 The upper high-level ontology

The upper high-level ontology, of which the most important concepts are shown in Figure 2.10, describes general classes, relations and axioms. An upper ontology, also known as a top-level ontology or foundation ontology, is an ontology which describes very general concepts that are the same across all knowledge domains [49]. Common examples are upper ontologies modeling time, e.g., the SWRLTemporalOntology [45], language, e.g., WordNet [50], common sense knowledge, e.g., Cyc [51], the Upper Mapping and Binding Exchange Layer (UMBEL) [52] and DOLCE [53], three-dimensional objects and processes across time, e.g., the Basic Formal Ontology (BFO) [54] and the General Formal Ontology (GFO) [55], or a combination of all of these, e.g., the Suggested Upper Merged Ontology (SUMO) [56]. Upper ontologies have however received a lot of critique. It is difficult to capture the concepts in such a way that they can easily be re-used across different domains without modification. Specific domains often require small alterations to the concepts defined in the upper ontologies. Upper ontologies also tend to define a broad spectrum of concepts as they are meant to be used across several domains. Only a small set of these concepts are then usable for the specific domain ontologies. Sometimes the concepts are too generic to be usefully re-used.

A lot of the concepts defined in the upper ontologies clashed with concepts defined in the ontologies we already wanted to import, e.g., with concepts in the OWL-S or WSN ontology. However, it is important to model time in the ontology, e.g., tasks that need to be performed before a certain point in time, for logging purposes or work processes that are adjusted according to the time of day. These concepts were unavailable in the ontologies we already wanted to import. Consequently, we chose to import the SWRLTemporalOntology [45]. It defines a temporal model that can be used to model complex interval-based temporal information in OWL ontologies. A library of SWRL built-ins to perform temporal operations and reasoning on information described using this ontology [57] is also available. The imported concepts are preceded with the temporal namespace in Figure 2.10.

The upper high-level ontology, developed in this research, extends the SWRLTemporalOntology ontology with additional concepts to model the current temporal context, e.g., the current season, the current shift or holidays. Most importantly this ontology enables that data can be related with a unique ID. All the other high-level ontologies import the upper high-level ontology and define all their concepts as subtypes of the `ExtendedProposition` concept.

2.3.1.2 The sensor high-level ontology

As the ambient intelligent care room of the future will contain numerous sensors and actuators to support caregivers and care receivers in their everyday activities,

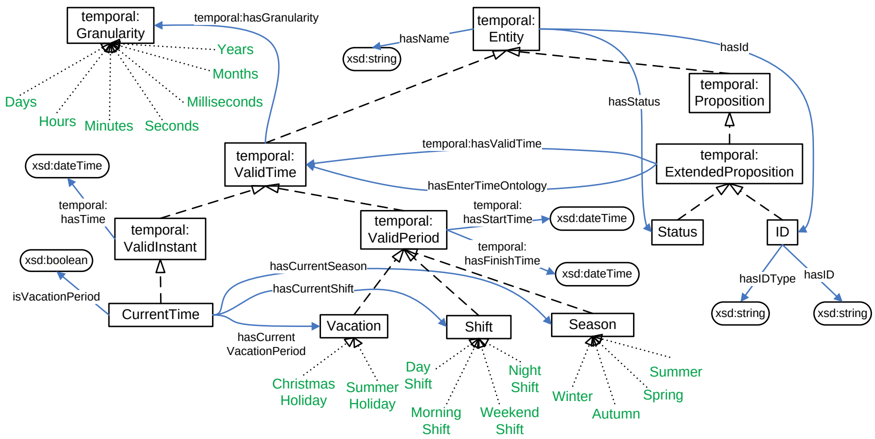


Figure 2.10: Prevalent concepts of the upper high-level ontology

it is important to model the information delivered by these devices in the ontology. There exist a whole plethora of ontologies to describe sensors and sensor networks. A comprehensive list can be found in Section 4 of the W3C Semantic Sensor Network XG Final Report [58] of the W3C Semantic Sensor Network Incubator group. Within the iMinds DEUS Project [59] a Wireless Sensor Network (WSN) ontology was developed [43]. The most important concepts of this ontology are visualized in Figure 2.11, preceded with the *wsn* namespace. The model contains a `System` concept which models a system and its components. The ontology allows interpreting and attaching meaning to the data values monitored by the sensors. For this the ontology uses an observation pattern. A data value monitored by a system is modeled in the ontology as an `Observation`. Rules added to the ontology allow detecting specific phenomena in these observations, e.g., the humidity is zero or the new temperature observation deviates more than 10 degrees from the last. These phenomena are modeled as `Symptom` concepts. Classification axioms allow reclassifying these symptoms as `Fault` concepts, e.g. the temperature sensor is possibly faulty or the humidity in the room is too low. Each fault can then again be reclassified as a `Solution` concept, e.g., do not use that temperature value or turn on the humidifier in the room.

The sensor high-level ontology extends this WSN ontology with systems, sensors and actuators, e.g., nurse call buttons or fall detection systems, and their associated observations, faults and solutions that play an important role in continuous care settings. Some representative examples are shown in Figure 2.11.

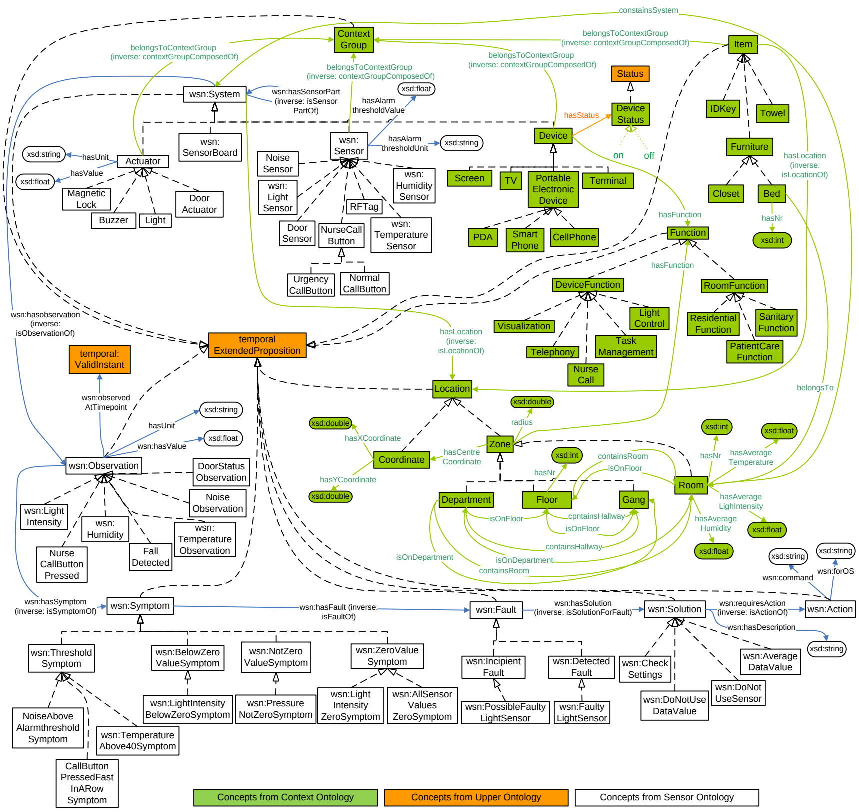


Figure 2.11: Most important concepts of the sensor & context high-level ontologies

2.3.1.3 The context high-level ontology

The context high-level ontology models the contextual environment information. The prevalent classes of this ontology are visualized in Figure 2.11. The most important concept is the `ContextGroup`. A context group is a logical grouping of entities that belong together, e.g., a care receiver with all his or her devices, sensors, actuators, room, bed, equipment and items. This is a dynamical concept which is defined through rules. The composition of a context group constantly changes based on information in the ontology, e.g., the location or status of the people, devices and equipment.

This ontology also contains all the information related to localization. A `Location` can be a `Coordinate` or a `Zone`. A coordinate is fully defined by its x- and y-coordinates. A zone is fully defined by its centre coordinate and a radius. Examples of zones are rooms, hallways, departments and floors. This ontology also introduces another subtype of system, namely a `Device`. A device is a logical grouping of sensors and actuators. A device is associated with a status, e.g., on or off. Devices and zones can be associated with their `Function` or purpose.

2.3.1.4 The profile high-level ontology

The profile high-level ontology, of which the most important concepts are visualized in Figure 2.12, models the profile information about staff members and care receivers that was indicated by the stakeholders in the different co-design workshops as important to take into account when optimizing continuous care processes. This model associates each `Person` with a `Profile`. A `Profile` consists of a `Basic` and a `Risk Profile`. The basic profile models the important administrative, e.g., phone numbers and birth date, biological, e.g., sex, psychological, e.g., aggressive or impatient, and sociological information, e.g., nationality and language, about a person. This information needs to be inputted into the system or extracted from documents, e.g., the medical file of a care receiver. On the other hand, the risk profile is defined by classification axioms and rules. This allows a reasoner to automatically obtain the risk profile of the care receiver by reasoning on the information in the basic profile.

The profile high-level ontology also contains concepts and classification rules to model the `Trust Relationship` between two people. Three kinds of trust relationships are differentiated, namely `Family Relationship`, `Personal Relationship` and `Therapeutic Relationship`. The first is used to indicate that two people are related. The second is used to indicate that two people have some kind of personal bond, i.e., a friendship. The third indicates that a trust relationship has been established between two people in the context of this person's care/health. Each trust relationship can also be associated with a degree that expresses the strength of the relationship. This allows that the trust relationship is taken into account according different degrees of specificity by different applications. For some applications it might be enough that there exists some kind of trust relationship between two people, e.g., a physician has a therapeutic relationship with a care receiver and is therefore allowed to access the medical files of this person. For other applications the degree of the trust relationship might also be important, e.g., in the Dominiek Savio Institute each resident has a personal assistant with whom they might have a personal bond of high degree, next to the therapeutic relationship the resident has with this caregiver. Finally, multiple types of trust relationships can be simultaneously defined between two people. For example, consider a patient who has a bad trust relationship with a close relative, e.g., a father-son relationship of first degree. In this case, this patient has a `Family Relationship` with this person of high degree, but a `Personal Relationship` of very low degree.

Finally, the profile high-level ontology also associates the `Person` concept with a lot of concepts from the already discussed ontologies to indicate, e.g., the location of the person, which items and devices he/she owns and the context group this person belongs to. A person can also have a status, e.g., free or busy.

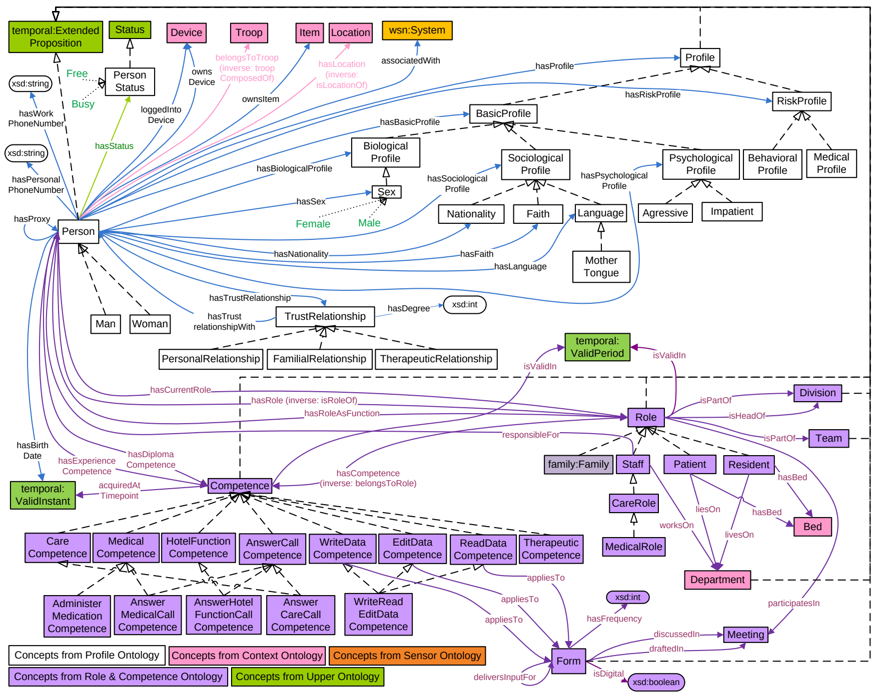


Figure 2.12: Most important concepts of the profile and role & competence high-level ontologies

2.3.1.5 The role & competence high-level ontology

To support caregivers in their tasks it is important to know the competences and the roles of the different staff members and care receivers. The role & competence high-level ontology, of which the most important concepts are visualized in Figure 2.12, models the different roles which can occur in a continuous care setting.

A lot of research has been done on how roles should be represented in knowledge representation models [60–63]. In ontology engineering a distinction is often made between Entities, i.e., things that are, Events, i.e., things that happen, and Roles, i.e., things that are, but only in the context of things that happen. There are 2 criteria to distinguish roles from entities [64]:

- a role is 'founded', i.e., it is defined in terms of relationships to other things, and
- a role lacks 'semantic rigidity', i.e., its existence is not tied to its class.

By this definition, a `Person` is thus an `Entity` and not a `Role`. Roles also have their own characteristics [65]:

- Roles are created and destroyed dynamically. As an entity only has a role in context of an event, the role is created when the event begins. When the entity no longer takes part in the event, the role may cease to exist.
- A role can be transferred between entities. For example, the role of department head can be transferred from one person to another. Some of the characteristics of the role are transferred without change, while others must be adapted.
- An entity may play different roles simultaneously.
- Entities of unrelated types can play the same role.

Three basic approaches for representing roles are commonly used in literature and are presented below:

- A role can be represented as a label assigned to a participant in an event [65]. This approach is simple, but it does not distinguish roles from entities and makes it difficult to add characteristics to roles. It thus fails to meet the third characteristic of roles.
- The second approach differentiates roles from entities [66, 67]. These two concepts are then however combined into one single hierarchy. This combination can be done in two ways:
  - The roles can be subtypes of entities. This is however problematic when entities of different types can play the same role (4th characteristic).
  - The roles are super-types of entities. For example, `Person` would then be a subtype of `Caregiver`. This would however mean that each person is a caregiver. It also fails to meet the first characteristic.
- The third approach [60] represents a role as an 'adjunct instance' of an entity, which is a distinct instance of a role class that is coupled with the instance of an entity. The role instance does not exist independent of that entity.

The third approach meets all the characteristics of roles and is also the approach that is extended in this research. In our representation, roles are types independent of people/entities with two separate hierarchies. Both are subtypes from the general `ExtendedProposition` concept. An instance of a role is played by an instance of an entity. Thus, every instance of a role exists along with an instance of an entity.

Our representation extends further on this representation by also including `Competence` concepts. Each role is defined by its competences through classification axioms, e.g., the `Doctor` is defined as a role which has all the medical competences. Each person is associated with competences and roles through five relationships:

- `hasFunction`: primary role of this person, i.e., the role for which this person was primarily hired.
- `hasRole`: models all the roles this person can have.
- `hasCurrentRole`: role the person is currently fulfilling within the care setting. If this relation is not instantiated, it is assumed that the current role of the person is his or her function.
- `hasDiplomaCompetence`: extra competences this person has acquired by following courses, e.g., a caregiver who is trained to perform some medical tasks.
- `hasExperienceCompetence`: extra competences this person has acquired through experience.

Roles can also have properties in our representation, e.g., the department a person works on when he/she has a specific role. In order to retrieve values of properties that belong to a role, first the role from the entity with which it is composed needs to be retrieved, and then the values of properties from the role can be obtained. Defining each role by its competences through axioms allows writing algorithms that find the most appropriate staff members to fulfill a task based on the required competences.

Little previous work was found on modeling concrete roles and competences which occur within the continuous care domain. However, as we also want to take into account the family relationships of the care receivers, the Family SWRL Ontology [68] was reused. This ontology also contains rules to derive family relationships from other defined relationships. However, this ontology declares the `Family` concept as a subtype of the `Person` concept. This contradicts with our choice to separate the definition of roles and entities in two hierarchies. Therefore, this ontology was not directly imported, but slightly adapted such that the `Family` concept is defined as a subtype of the `Role` concept. This is visualized in Figure 2.12 by the concept preceded with the *family* namespace.

2.3.1.6 The task high-level ontology

Since the goal of this research is to support caregivers and care receivers in their daily activities and tasks, it is important to model these process workflows in the ontology. A wide range of ontologies exist to model processes, mainly focused on business process management. A lot of these ontologies are not expressed in OWL [53, 69–72]. These cannot be directly imported in our ontology without translation to OWL. Translating the ontologies would break the links with the original ontology, which means we would have to adapt our ontology manually everytime the original ontology is changed. This goes against the linked open data (LOD) paradigm [73]. Good alternative OWL ontologies exist to express process workflows. The most well-known is the Business Management Ontology [74], which represents an integrated information model, which helps to better align IT with business and allows constructing process flows. However, it is tuned towards fostering the communication between business analysts and software developers. For this, this ontology can be compared to how UML is traditionally used to support this communication. It is limited in the availability of semantic concepts that allow constructing process workflows automatically from a set of specified tasks and the available input and desired output. It does allow specifying links between tasks and roles and how they map on each other. However, only high-level concepts of this role definition can be re-used as most of the specified roles are not applicable to the healthcare domain. Moreover, as explained in the previous section, competences should be considered separately from roles in this research, as not every competence in a healthcare setting is covered by specific roles and people can follow additional courses to gain extra competences which are not covered by the roles they have. Tasks should thus be mapped on the competences needed to perform the task instead of on specific roles. Another well-known ontology is OWL-S [44], which is an ontology for describing Semantic Web Services. It enables users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints. The OWL-S ontology has three main parts: the service profile, the grounding and the process model. The latter is the most important for our goal. The process model describes how a client can interact with the service. This description includes the sets of inputs, outputs, pre-conditions and results of the service execution. Consequently, it allows to describe how processes can be mapped on each other based on their inputs and outputs, which conditions need to be fulfilled to execute the process and which effect the execution of the process has on the environment and the context. This allows automatically mapping the effects and outputs of one task on the inputs and conditions of another task. This way workflows can be constructed that start from particular input and context and reach a specified effect and result by combining various tasks. It does not define the mapping of processes on competences or roles. However, this can easily be defined by ourselves based on literature as specified in the previous section. For these reasons, we chose to import the OWL-S ontology to model tasks.

The most prevalent classes of the OWL-S Process ontology and their relations are visualized in Figure 2.13, preceded by the *owls* namespace. The central class is the `Process` concept. A process can generate and return some new information and produces a change in the environment based on the information it is given and the context. This is described by the `Input` and `Output` concepts and accompanying `hasInput`, `hasOutput`, `hasPrecondition` and `hasEffect` relations. For a process to be able to execute, all its preconditions need to be fulfilled. The `Result` of the process is thus the combination of its output and its effect. A process can be composed of several other processes. How these processes are combined is expressed by the `CompositeProcess` and `ControlConstruct` concepts.

The High-Level Task ontology extends this OWL-S Process ontology, as shown in Figure 2.13. It is modeled that the precondition of a process can depend on the value of a sensor and that the effect can be the status change of an actuator through the `usedIn` and `controls` relations. The `Task` concept is introduced, which is equal to a `Process`, but is further divided into planned and unplanned tasks. Each task has also an associated `Status`, e.g., `Assigned` or `Finished`, and `Priority`. From the observations and workshops it was clear that the stakeholders preferred three levels of priority, namely very urgent, urgent and normal. This `Task` concept can now be used to model the various continuous care tasks. Each task is defined by the `Competences` which are needed to execute this task. Also the location at which this task is preferably executed can be indicated.

Consider, for example, the task of assigning a person to a call or care request. A `Call` is modeled as an unplanned task. There are four possible types of calls. A `Normal Call` is a call made by a `Person`, i.e., care receiver. An `Assistance Call` is a call made by a staff member who requests the assistance of another staff member. A `Context Call` is a call which is automatically launched as a consequence of an `Observation` made by a `System`. An `Urgency Call` is a special type of call which is used in emergency situations. A call can also be associated with a `Reason`, i.e., medical, e.g., medication needed, care, e.g., giving a person a bath, and hotel, e.g., fetching a glass of water, are also modeled. This allows that a call is reclassified based on its reason. For each type of a call it can then be specified which competences are needed to handle a call of this type, e.g., `Medical Calls` can only be handled by a person who has medical competences. For each type of call, its preconditions, e.g., a care receiver pushes a call button, its input, e.g., the person who pushed the button or the location of the button, its output, i.e., the assigned staff members and its effects, e.g., the assigned staff members' portable phone rings, are modeled by using concepts from the OWL-S Process ontology.

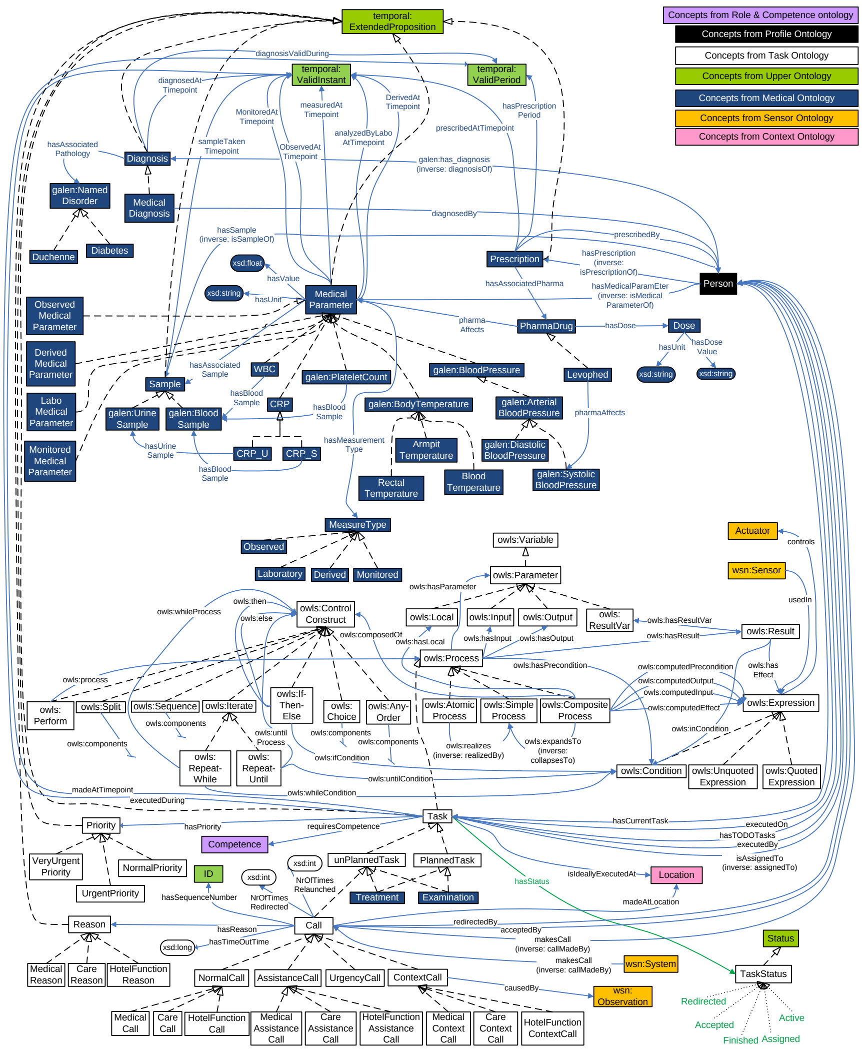


Figure 2.13: Prevalent concepts of the task & medical high-level ontologies

2.3.1.7 The medical high-level ontology

As mentioned in Section 2.1.2, a wide range of ontologies exist about the eHealth domain. For this research, we are mainly interested in representing medical knowledge that influences the way care is organized for this care receiver, e.g., diabetes (influence on food consumption), heart disease, trouble breathing or muscle paralysis (higher need for care). The Galen Common Reference Model is a model of clinical categories plus sufficient information about those categories to allow them to be classified automatically. The Galen Common Reference Model especially avoids adding too many axioms that constrain the possible interpretations of a concept, unless there is very wide agreement about the constraint, e.g., an ulcer located in the stomach is a stomach ulcer. Therefore it was decided to use the Galen Common Reference model as this model could most easily be extended without contradicting with knowledge already contained in the model.

As shown in Figure 2.13, the medical high-level ontology adds axioms and constraints to the imported Galen Common Reference Model that express relations between this medical knowledge and additional medical concepts needed for the continuous care domain. For example, the `Named Disorder` concept is connected with the `Diagnosis` concept from the Medical High-Level Ontology through the `hasAssociatedPathology` relation. The concepts from the Galen Common Reference Model in Figure 2.13 are preceded with the *galen* namespace.

The medical high-level ontology associates each `Person` with his or her `Medical Parameters`. How this parameter was measured is indicated by the `hasMeasurementType` relationship. The parameter is either observed by a staff member, measured by monitoring equipment, derived from other parameters or has been processed in a laboratory, e.g., by analyzing a urine sample. Each parameter can thus also be associated with the sample it was derived from, e.g., a `Urine Sample`. Finally, a `Person` is also associated with his or her `Prescriptions` and `Diagnosis`. Each `Prescription` is associated with its prescribed `PharmaDrug` and `Dose`.

2.3.2 The low-level ontologies

Two low-level ontologies were designed, namely one focussing on care tuned towards residential care settings and one focussing on cure and thus tuned towards hospital settings. The low-level care and cure ontologies consists of five and four sub-domains respectively, namely representing information pertaining to:

- roles for the care & cure domain,
- competences for the care & cure domain,
- forms used within the care & cure domain,
- care & cure tasks,
- profile information specific for the care domain, and
- context location information specific for the cure domain.

The import schema of all the ontologies that make up the low-level care (indicated in green) and cure (indicated in blue) ontologies and how they relate to the ontologies of the high-level ontology is visualized in Figure 2.9. Some example concepts from the low-level ontologies are shown in Figures 2.14 and 2.15. All these classes and relations are derived from the mind maps and document flows, see Section 2.2.2, constructed based on the observations and on the results from the decision-making workshops, see Section 2.2.5. Additionally, the medical competences were derived from an official document, namely 'Inventory of Nursing Acts', which details all the competences that nurses can execute according to their diploma, e.g., A, B or C, as stipulated by the Belgian Government. The low-level cure & care ontologies contain some concepts with the same name. These were not included in the high-level ontology as they are not used with exactly the same meaning, e.g., documents with a different purpose, and roles with the same name that have different competences and responsibilities within the two settings.

The low-level cure context ontology extends the concepts of the high-level context ontology, see Section 2.3.1.3, with subclasses that model typical departments, e.g., the `Emergency Department` or `Radiology`, rooms, e.g., `Nurse Post` or `Reception`, and hallways, e.g., `Stroke Unit`, found within hospital settings. The goal of the low-level ontologies is to define generic algorithms and capturing knowledge that can be used across the various cure and care settings respectively. Consequently, the low-level context ontology was only constructed for the cure domain as hospitals, in contrast to residential care settings, typically have a very common structure with the same type of rooms, departments and hallways found across multiple hospitals. Moreover, specific competences, tasks and patient profiles are often associated with these different types of spaces. These associations can be defined in the ontology. As no such common structure exists amongst the care settings, such associations cannot be defined in a generic manner.

The low-level care & cure document ontologies extend the `Form` and `Meeting` classes of the high-level role & competence ontology, see Section 2.3.1.5. These `Form` subclasses represent common document formats which are used within residential care and hospital settings and how they are related to each other. Additionally, this ontology defines the meetings which typically take place within these settings. It is defined which forms are used or produced within a meeting.

The low-level care & cure competence ontologies extend the `Competence` class of the high-level role & competence ontology, see Section 2.3.1.5, with classes that represent the competences which are needed to execute all the tasks and daily activities within residential care settings and hospitals and how they are related to each other. `Medical Competences` represent the competences which are needed to medically care for the care receiver, i.e., cure this person, and are usually fulfilled by staff members with a medical diploma, i.e., nurses and doctors. All the competences related to caring for the care receivers, e.g., `Washing` or `Dressing a Patient`, are modeled as subclasses of the `Care Competence` class. Competences which pertain to performing hotel tasks, e.g., changing or

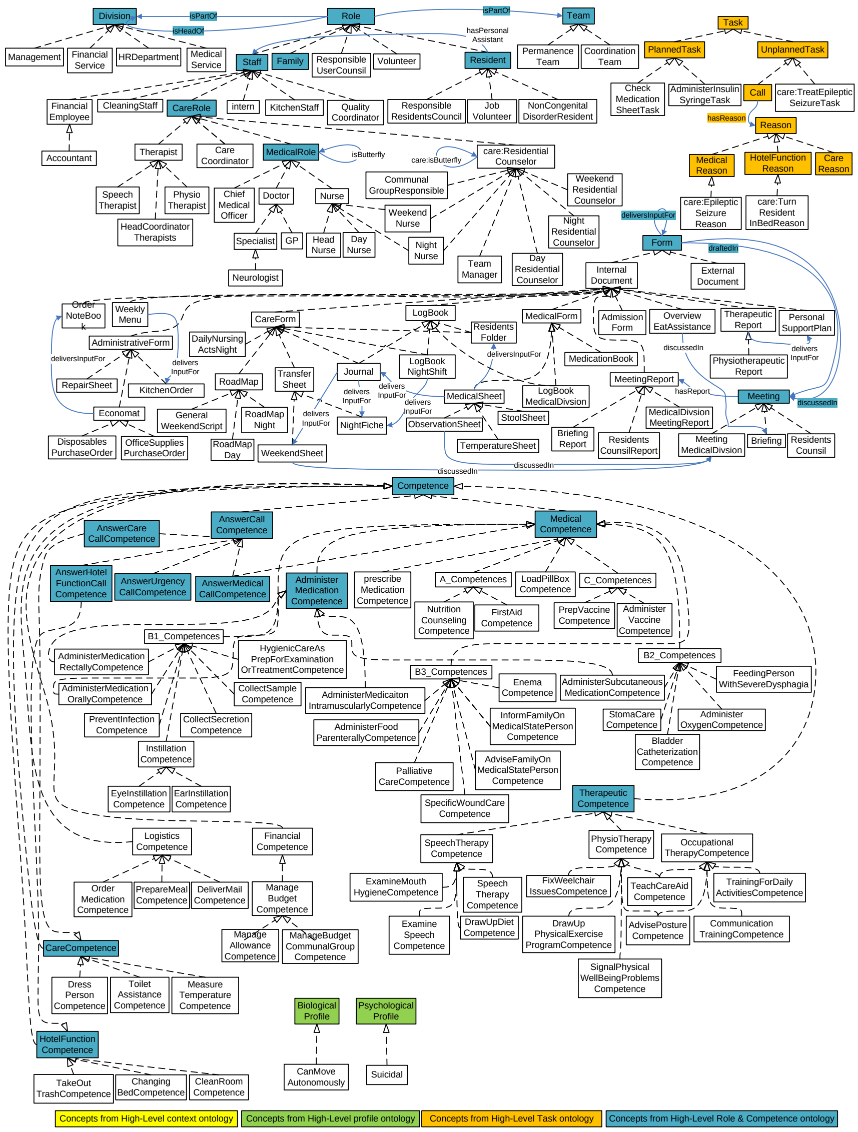


Figure 2.14: Example concepts of the low-level care ontology

making the bed or cleaning, are modeled as subclasses of the `Hotel Function Competence` class. The `Therapeutic Competence` class is the superclass of all the competences needed to draft and follow-up a specific therapy for a care receiver. These competences are grouped according to the different therapeutic branches, i.e., `Logopaedic`, `Physiotherapeutic`, `Dietary` and `Occu-`

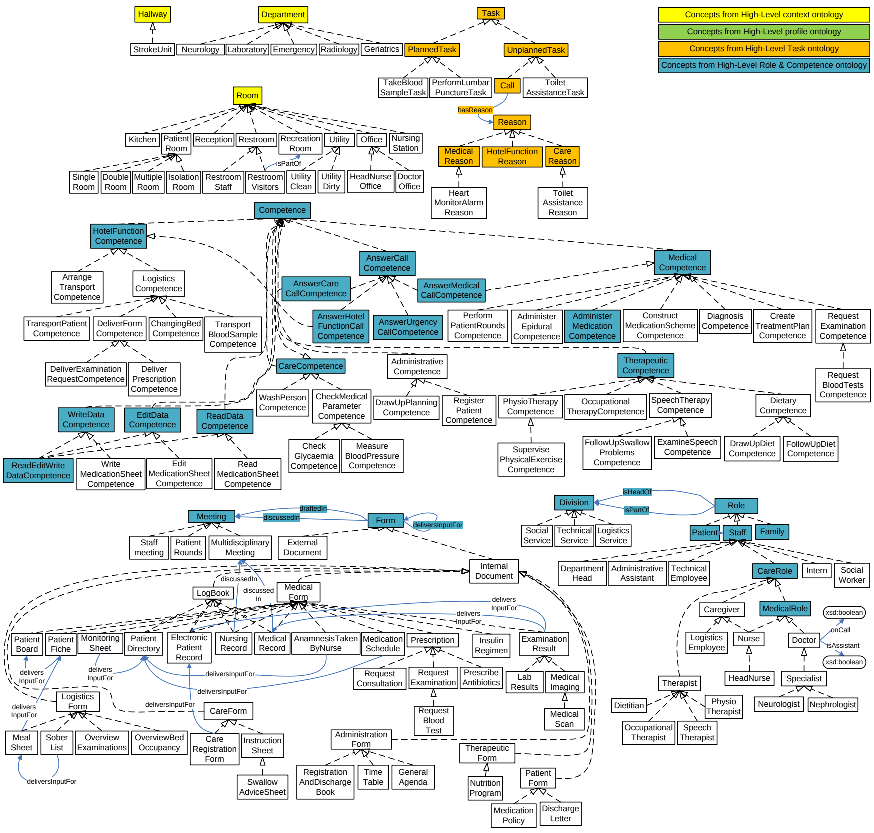


Figure 2.15: Example concepts of the low-level cure ontology

pational Therapy Competence. All the competences related to managing the logistics of the care setting, e.g., preparing meals, transporting care receivers or ordering medication, are grouped under the Logistic Competence class. Competences are also modeled pertaining to managing the finances of the residential care setting of specific residents as subclasses of the Financial Competence concept. Finally, competences are defined pertaining to reading, writing and adapting documents. Subclasses of the Edit, Read and Write Data Competence classes for each Document defined in the low-level care document ontology can be created. These subclasses then represent the competence to write, edit or read this specific document. These subclasses are connected through the appliesTo relation with the specific document they pertain to. One example of these subclasses is shown in Figure 2.15, namely the competences to read, write or edit a Medication Sheet.

The low-level cure & care role ontologies extend the Role, Team and Division classes of the high-level role & competence ontology, see Section 2.3.1.5. The Role subclasses represent common roles which people occupy within the respective settings and how they are related to each other. A distinction is made between Staff Members, Residents or Patients and Family. In the care settings, Volunteers and the Responsible of the User Council are also modeled. The Family roles represent the family of the care receiver, as has been already explained in Section 2.3.1.5. The Residents represents the people who are being cared for at the residential care setting. Further subclasses can be defined based on the specific illnesses treated at the care setting, e.g., Not-Congenital Disorder Resident, or specific tasks that the resident is allowed to perform, e.g., Job Volunteer or Responsible of the Residents' Council. Staff Members represent all the employees. An important subclass is the Care Role class, which models all the roles which handle caring for a person. A distinction is made between Therapists, Medical Roles, Residential Counselor (care) or Caretakers (cure). The Therapist, e.g., Speech Therapist or Physiotherapist, Medical Role, e.g., Doctor or Nurse, Residential Counselor, e.g., Day Residential Counselor or Team Responsible, and Caretaker, e.g., Logistic Employee, classes also have further subclasses to model specific responsibilities and competences. In the low-level care ontology, an important distinction is made between Residential Counselors and Nurses who work during the Day, the Night and the Weekend as they do not have same competences and tasks, e.g., access to different documents or the weekend and night nurses are expected to also perform caring tasks and not only medical tasks. This ontology also includes some additional caring roles, such as the Department Head or the Care Coordinator, which can be occupied by residential counselors or nurses. This decision depends on the specific situation and care residence. Therefore, these roles are modeled as direct subclasses of the Care Role class. Furthermore, it also indicated through the isButterfly relationship whether the Medical Role or Residential Counselor is a butterfly. A butterfly is a person who is not connected to a specific department, but can work on different departments depending on where staff is needed at the moment. Next to the caring roles a lot of other roles are needed at both the residential care setting , e.g., Social Worker or Kitchen Staff, and hospital, e.g., Cleaning Team or Administrative Employee. These roles are modeled as direct subclasses of the Staff Member class. A final important Staff Member role to notice is the Intern role. Each role can essentially be occupied as an intern, e.g., a nurse intern or a residential counselor intern. However, the tasks that interns can perform are very different and often more limited. They also often need supervision to perform certain tasks and their competences and responsibilities often change depending on their experience level. Therefore, a separate class was created for the Intern role and the hasExperienceCompetence and hasDiplomaCompetence relations, defined in the high-level role & competence ontology can be used to specify their competences and relations depending on the specific situation and context.

Each care & cure role is then further defined by mapping it on the appropriate care & cure competences, which are modeled in the low-level cure & care competence ontologies, through classification axioms that use the hasFunction, hasRole, hasCurrentRole, hasDiplomaCompetence and hasExperienceCompetence relationships defined in the high-level role & competence ontology.

Additionally, these low-level cure & care role ontologies define the divisions and teams which typically are present within the care & cure domain. It is defined which roles are part of a certain team or service and which role is the head of a service. For example, each person who has the role Social Worker is automatically part of the Social Service and each Logistics Employee is part of the Logistics Service. The low-level cure role ontology also defines a new relation, namely isHeadOf, which indicates which person with which role is the head of a certain Department, as defined in the low-level cure context ontology.

The low-level care profile ontology, extends the Profile class of the high-level profile ontology, see Section 2.3.1.4) with subclasses that represent common traits which residents display at a residential care setting, e.g., being able to move independently or being suicidal.

The low-level care & cure task ontologies extend the Task and Reason classes of the high-level task ontology, see Section 2.3.1.6, with new subclasses. The Task subclasses represent common tasks which are performed within residential care settings and hospitals and how they are related to each other. By using the isInNeedOfCompetence, defined in the high-level task ontology, these care tasks can then be mapped on the care & cure competences, defined in the low-level care & cure competence ontologies, needed to perform the task. By using the other relations defined in the high-level task ontology the different care tasks can be related to each other. For example, the task Administer Insulin Syringe can only be performed by people who occupy a role that has the competence Administer Intravenous Medication. Moreover, this task is a Composite Process, which is composedOf a Sequence, with as components first the Check Medication Sheet task and second the Administer Syringe task. For the first task, the Read Medication Sheet Competence is needed.

2.3.3 The rule-based algorithms

On top of this ontology rule-based algorithms are developed that contain the algorithms to optimize and automate continuous care. These rules infer new knowledge based on the data available in the ontology. Efficient and fast notifications made by these algorithms allow appropriate actions to be taken by the staff. An example of a work process that can be optimized this way is the assignment of caregivers to calls or care requests for which part of the decision tree is shown in Figure 2.7. For example, the following rule assigns a staff member with medical competences to an urgent medical request from a care receiver:

```
MedicalCall(?call) ∧ hasCurrentRole(?p,?ro) ∧ Staff(ro) ∧
hasCompetence(?ro,?c) ∧ MedicalCompetence(?c) ∧
hasPriority(?call,?prior) ∧ UrgentPriority(?prior)
⇒ assignedTo(?call,?p)
```

2.4 Discussion

In Table 2.1 we compare the various methods on different parameters. The first parameter concerns the number of participants. We experienced that a large group of participants is beneficial for explorative workshops like workshop type 1. A large group of stakeholders were sensitized and got their first experience with ontologies and the project. Many of these participants were also involved in the other workshops. Therefore, this workshop type was a good way to create awareness and advocates in the stakeholders group. But, for instance, workshop type 2 may have benefited from a smaller group of participants. The issues raised during the discussions varied widely and the researchers struggled to keep focus in the conversation. A reduced number of participants may have made it easier to moderate. The same thing can be said about workshop type 3a and 3b. Workshop type 3b resulted in more targeted feedback thanks to the smaller group size.

The second parameter concerns preparation time, which also differed widely between workshops. Part of this can be explained by the experience the researchers gained throughout the research process in finding ways to link user feedback to the ontology and the prototype. For instance, workshop type 5 was more efficiently prepared thanks to the experience gained in the other workshop types. The workshop type 2, however, took a lot of preparation time given the novelty of the objective of the workshop for the researchers at that time. With regard to execution time of the workshop type, clearly the observations at the start of the project and the embodied system use in workshop type 5 required the biggest efforts of the researchers, which mainly has to do with the methods (contextual inquiry and small group role-play) chosen in those workshop types.

The final parameter in Table 2.1 is the main impact of the workshop type on the ontology creation process. We have mapped our workshop methods to three of the five steps Pinto and Martins [14] defined for ontology development, namely: specification, conceptualization, formalization, implementation and maintenance. However, this classification should not be interpreted too strictly, since in reality every workshop type resulted in some feedback for every stage in the ontology engineering process.

The domain experts were not included in the implementation stage, since this concerns the translation of models as, e.g., the decision tree, into the language of the ontology. This is unrelated to domain expertise, and since we did not want to overburden the domain experts, we did not involve them in this phase. The final step of Pinto & Martins is also excluded from Table 2.1, since in our opinion the step could in practice not be distinguished from the other steps in the process. Just as the future care concepts were continuously evaluated by new iterations of the scenario in the innovation binder, the continuous care ontology was permanently maintained and adjusted in every step of the ontology creation. While Pinto and Martins reserve this phase for the last step in the ontology creation, in our case, we started maintaining and adjusting the ontology after every workshop type.

However, in respect to Table 2.1, we also wish to express that, while we found it important to make this comparison of the different workshops, we also wish to stress its relativity by underlining that the parameters were also determined by the experience the researchers involved had at the time and the sequential order of the workshops. It is therefore difficult to determine which workshop was most insightful, although it is clear that some workshop types were more successful in reaching their objectives than others, i.e., workshop type 3 (decision-making) and 5 (embodied system use).

		Observations	WS1 Intro ontologies	WS2 Role-play	W3 a + b Decision-making	WS4 Concept evaluation	WS5 Embodied System use
Number of participants		+50	22	18	a) 16 b) 4	14	7 groups of 2 or 3 people
Time	Preparation	Medium	Long	Very long	Short	Medium	Short
	Duration	3 Weeks	1/2 day	1 day	2 hrs.	Half a day	2.5 hrs./session
Main impact	Specification	x	x				
	Conceptualization			x			
	Formalization				x	x	x

Table 2.1: Comparing observations and the various workshop types

In this research project, a prototype was developed to test the ontology and ensure its cross-institutional validity. Two comments can be given on this method for testing the cross-institutional validity of the ontology. First, while the two workshops that focused on the ontology resulted in a lot of feedback on the prototype, it was unclear to what extent this method was also successful for validating the ontology. The lack of feedback on the ontology can be understood as a validation of the ontology, but it can also mean that the method is not suited for ontology validation. Moreover, while the prototype covered some elements of the ontology, it did not include all of them. The question remains how this method can be used to validate all elements in the ontology. Given the high time investment needed not only for organizing the workshops, but also for development and implementation of the prototype, it seems unrealistic to develop several prototypes in order to cover every element in the ontology. Additional methods should be thought out for cross-institutional ontology validation. We are currently looking into methods and tools [75] which allow automatically translating an ontology into a natural language description of the concepts covered in the ontology. Users could read this generated story and identify incomplete or incorrect sentences. This would be less time-consuming for the involved ontology engineers and social scientists than organizing a large number of workshops of type 5 as this task can be performed by a large number of targeted end-users simultaneously without (constant) support. However, due to the large amount of text generated from a large ontology, rapid or less-focussed reading of the users could become a problem. To counter this, it would be interesting to use some form of gamification to increase the attention span and thoroughness of the evaluation. We are also looking into available tools to formally evaluate an ontology, such as OntoClean [76] and AEON [77]. Based on these user remarks and the results of the tools, additional workshops or prototypes could be made to fill these gaps. Ideally the three approaches, e.g., workshops of type 5, the reading exercise and formal evaluation tools, complement each other. Workshops could be organized for the most difficult or discussed parts of the ontology, while the reading exercise and formal tools could be used to evaluate and validate the other parts and resolve the small nuances and inaccurate statements. Moreover, it could also diminish the problem of rapid reading.

The innovation binder took a central position in the project, not only as a project result, but also as a project management tool. It served as a boundary object between all project partners and created a common perspective that was supported by all stakeholders involved in the project. Every step of the future scenario that was included in the innovation binder got translated into implications for every project partner. While this is a useful exercise, it is also an extensive one. Every insight gained during the workshops might result into changes in the scenario, which may then again have to be translated in new implications for the project partners. These continuous iterations on the scenario and its implications require a high engagement of all project partners and can only work when meetings are organized on a regular basis with a small group of engaged people.

An important challenge was to make the innovation binder both a shared vision and a tool. The information in the innovation binder had to be somewhat ‘visionary’, but at the same time could not be too high level and had to be translatable into concrete requirements. At one time in the process, i.e., after workshop 2: Scenario role-play, it actually seemed that the innovation binder got too high level, missed its practical use and did not deliver the expected results. Making the scenario somewhat less future oriented and more concretely linked to current practices helped to increase its usability.

2.5 Conclusion: Lessons learned

In this paper, we have described the steps we have taken in order to involve stakeholders and targeted end-users in the development of a continuous care ontology and continuous care concepts. To that aim, we have created a stakeholder group, did observations in both a care and a cure setting and organized a series of workshops that aimed to involve domain experts from the beginning, but at the same time did not require efforts that are too demanding on their part. The methodology actively involves ontology engineers, social scientists and stakeholders, i.e., nurses, caregivers, care receivers, doctors and professionals working for the healthcare industry, in the ontology engineering process.

It should be stressed that while this project involved ontology engineers, implementers and social scientists who have a lot of experience involving users in design processes, the thought-out process was largely experimental and all researchers involved had to creatively use the methods they were familiar with in order to achieve the project's goals. While it is important to start from best practices with regard to the methods used, the researchers involved also had to be somewhat reckless at times in order to find a way to involve the domain experts' perspective in the creation of the ontology. We acknowledge that the process that we have followed might not be recommended at all times. Indeed, it is sometimes difficult to fully grasp the effectiveness of a workshop, or to differentiate whether a method is (un)suited or simply confirms the findings. We therefore hope that the process we have followed is inspiring and encourages other researchers to do similar experiments and adjust the methods we have used to their own needs. In the following paragraphs, the most important lessons learned during the co-creation of the ontology with the participatory methodology are discussed.

Creating common ground: A step that is elementary in the process is creating common ground with the stakeholders or target end-users involved in the workshops. In general, there was some discussion to what extent the users should understand the full complexity of the project and ontology when participating in a workshop. While reaching common ground was explicitly the aim of our first workshop type, we somewhat overlooked the importance of this aspect when involving a larger group of stakeholders in the fourth workshop type, and as a result this workshop resulted in discussions that were similar to the ones we had in the first workshop. As a consequence, a far more elaborate introduction was included in the fifth workshop type, with better results.

Elicitate out-of-the-box thinking: Reserving sufficient introductory time is not only important for creating common ground, but also to facilitate the participants to move from thinking about current practices to future practices in the workshops. At first, participants might be overwhelmed or amazed by the possibilities of a new system, or contrary fiercely opposed to the idea of installing automated processes. It takes some time to overcome those first ideas, get insight in the reasons for their initial enthusiasm or aversion and stimulate creative thinking that goes beyond current practice, but still takes into account the particularities of the environment and the users, which the ontology and the applications built on the ontology are targeting. It was evident that simply bringing the stakeholders together with stakeholders from other disciplines was not sufficient to help the participant make this transition and think out-of-the-box. To this aim, it is also important to provide the right tools. For instance, immersive tools such as the persona and situation cards used in workshop types 2 and 5 served as good elicitation tools as it took stakeholders out of there usual role and context. The movie used in workshop type 4 was on the contrary a more distant, detached experience and thus resulted in less creative thinking.

Getting the group size right: Finding the right group size is a challenge and depends on the aim of the workshop. While in workshop type 1, the large size of the group was not a disadvantage since the aim was mainly to create common ground, a large group did become a problem when in-depth feedback was wanted. The series of small-sized workshop type 5 where very labor intensive, however, here the participants were more critical and gave more detailed feedback. This of course had also to do with the fact that the workshop gave them a deeper experience of the system.

Reaching consensus: At times in the process, particularly after workshop type 2, the lack of consensus among the participants was worrying for the research team. However, workshop type 3 made clear that this lack of consensus is actually an indication that the ontology should consider not including the concept in the high-level ontology. If the interpretations of a concept are dispersed, it might be best to move it to a low-level ontology or exclude it.

Working in interdisciplinary teams: Throughout the whole process, both social scientists and ontology engineers were involved in the process. This mixture had as an advantage that both the user and technical perspective were at the forefront in all stages of the ontology construction. Not only during the development of the method and practicalities of the workshop all parties were present, but also during the workshops themselves and during analysis. This facilitated the translation of user findings in the ontology and the innovation binder. The innovation binder in particular proved to be a powerful tool in this project in bringing together future continuous care concepts, and user and technical requirements. However, the tool would not have been successful if not supported by all parties involved.

Connecting ontology engineers and stakeholders: One of the challenges during the workshops was the facilitation of the communication between the ontology engineers and stakeholders. It became apparent that bridges needed to be build between them. This was done in various ways. First, the ontology engineers took part in the observations to get an idea of the current work practices of the stakeholders. Second, bridges were also used during the workshops, e.g., the storyboard in workshop 2 or the decision tree in workshop 3. Finally, the resulting ontology and axioms were communicated with the stakeholders in an easily understandable format, e.g., document workflows, mind maps, graphs and decision trees. We noted that some ways of building bridges were more successful than others. Mutual interaction seems to be key. In Workshop 2 there was limited interaction between the ontology engineers and the other stakeholders. Ontology engineers observed the others' work but stakeholders tended to forget to comment on the conceptual graph under construction. The stakeholders became oblivious of the ontology engineers' task and ontology engineers could not get feedback on the model as it was being constructed.

Timing: One of the goals was to achieve user participation beyond collecting the requirements of the ontology in the specification phase, but without actually pushing them in the role of ontology engineers and thus overburdening them. The stakeholders are mainly involved through the workshops and observations. The observations comprised of 1 week in each continuous care setting, during which staff members were followed and interviewed. The workshops lasted 2 to 3 hours on average. The workshops were distributed amongst the available participants, while maintaining the group of each workshop representative of the different stakeholders in the domain. Participants often remarked that the sessions seemed too short to get to the bottom of the issue at hand. However, it was difficult to lengthen the workshops as participants often could not make themselves available that long. In line with a suggestion of one of the participants, a follow-up session was coupled with each workshop in which the output of the workshop and the achieved results are presented. This permits to briefly discuss pressing issues not handled during the workshop, to evaluate the output and to illustrate to the stakeholders that their input was taken into account and that is was time well spent. It was also concluded that while the described methodology limits the time that has to be invested by the stakeholders, it does require a large amount of time and effort from the social scientists and ontology engineers.

Learning by doing: A hands-on approach was used during the workshops, e.g., exercises in workshop 1, role-playing in workshop 2 and 5 and question-and-answer process in workshop 3. Participants were also stimulated to reflect on sometimes highly complex issues. It was found that participants much appreciated this approach of action and reflection. It allowed them to reflect on their current practices, enhanced their understanding of the topic and elicitated discussion.

Getting the language right: At the start of the project, the question was put forth in which language the ontology should be created: Dutch (the mother language of the stakeholders) or English. Clearly each option has benefits. Dutch would make it easier to let stakeholders evaluate the ontology. It does not require command of a foreign language and may more easily invoke intuitive understanding. Furthermore, words used within the domain of interest could be inserted directly into the ontology. An English ontology, however, would make it easier to reuse existing ontologies and to reach an international audience. Also, the search for an appropriate translation would force stakeholders to have a thorough discussion of what the concept means to them and facilitate a truly shared understanding. Ultimately, we chose to have the best of both 'ontologies' by making the ontology in English and annotating it in Dutch using the labels defined by SKOS [78], e.g., skos:preflabel and skos:altlabel.

In future research, we are exploring if and how the current ontology for institutionalized care and cure can be extended to non-institutionalized care or home care. This is the focus of O'CareCloudS, the follow-up project of the ACCIO project. Other prototypes, e.g., a task management system and a context-aware information provisioning and filtering system selecting the approprate device, are also being developed using this continuous care ontology to evaluatie its completeness and validity. We also will investigate whether this research process can be repeated by engineers who are less knowledgeable about ontologies. Ontology engineers are currently relatively rare and therefore it would be interesting to see to what extent our method is easily reproducible and what the thresholds are.

2.6 Addendum

It is difficult to pinpoint the exact impact of the user involvement on the resulting ontology. One could wonder if they could be replaced by a study group on best care practices. In Chapter 4 a very early version of the ontology is presented and discussed. This ontology was arrived at through discussions with the engineers working at Televic NV, i.e., developers and researchers of nurse call systems, and targeted end-users who occupy managerial positions in hospitals. The caregivers and nurses who used the nurse call system and handled care requests on a daily basis, were not involved. If we compare the ontology shown in Figures 4.3, 4.4 and 4.5 to the ontology presented in this paper, the differences are very notable. For example, people and roles were not differentiated, the information on patients was mainly reduced to risk factor profiles and the ontology puts a lot of emphasis on concepts and definitions, which are not really used on a day-to-day basis, e.g,, a whole plethora of different priority categories and deeming characteristics of patient such as spoken languages, faith and gender as import for nurse call assignment. These small nuances in the ontology, have an impact on the developed PoC, as can be noted by comparing the PoCs presented in Chapter 4 and Appendix A.

Acknowledgment

Part of this research was supported by the iMinds ICON Project ACCIO. This project is co-funded by IWT, iMinds and the following partners: Televic Healthcare NV, Boone NV, Dominiek Savio Instituut and In-Ham. F. Ongenae would like to thank the IWT for financial support through her Ph.D. grant. Several workshops were organized in the PRoF demo rooms located in Poperinge, Belgium. We thank all the participants in ACCIO for their valuable contribution to the project.

References

[1] M. Satyanarayanan. *Pervasive Computing: Vision and Challenges*. IEEE Personal Communications, 8(4):10–17, 2001.

[2] N. Roy, T. Gu, and S. K. Das. *Supporting pervasive computing applications with active context fusion and semantic context delivery*. Pervasive and Mobile Computing, 6(1):21–42, 2010.

[3] J.-C. Burgelman and Y. Punie. *Close encounters of a different kind: ambient intelligence in Europe*, pages 19–35. Springer-Verlag, Berlin-Heidelberg, 2006.

[4] Y. Punie. *The future of ambient intelligence in Europe: the need for more everyday life*. Comm Strat, 57:141–165, 2005.

[5] F. Ongenae, A. Ackaert, A. Jacobs, A. Veys, M. Van Gils, P. Verhoeve, and F. De Turck. *User-driven design of an ontology-based ambient-aware continuous care platform*. In Proc. of the 4th International Conference on Pervasive Computing Technologies for Healthcare, pages 1–4, Munich, Germany, 2010.

[6] M. Tentori, D. Segura, and J. Favela. *Monitoring hospital patients using ambient displays*, chapter VIII. Medical Information Science Reference, USA, 2009.

[7] A. Valls, K. Gibert, D. Sánchez, and M. Bateta. *Using ontologies for structuring organizational knowledge in Home Care assistance*. International Journal of Medical Informatics, 79(5):370–387, 2010.

[8] A. Gomez-Perez, O. Corcho, and M. Fernandez-Lopez. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag, London, UK, 2003.

[9] F. Paganelli and D. Giuli. *An ontology-based system for context-aware and configurable services to support home-based continuous care*. IEEE Transactions on Information Technology in Biomedicine, 15(2):324–333, 2011.

[10] V. F. S. Fook, S. C. Tay, M. Jayachandran, J. Biswas, and D. Zhang. *An ontology-based context model in monitoring and handling agitation behaviour for persons with dementia*. In Proc. of the 4th IEEE International Conference on Pervasive Computing and Commununications Workshops (PERCOMW), pages 560–564, Pisa, Italy, 2006.

[11] D. Zhang, Z. Yu, and C. Y. Chin. *Context-aware infrastructure for personalized healthcare*. Studies in Health Technology and Informatics, 117:154–163, 2005.

[12] E. J. Ko, H. J. Lee, and J. W. Lee. *Ontology-based context modeling and reasoning for U-HealthCare*. Transactions on Information and Systems, E(90)-D(8):1262–1270, 2007.

[13] *The ACCIO Project*. http://www.iminds.be/en/research/overview-projects/p/detail/accio-2, 2012.

[14] H. Pinto and J. Martins. *Ontologies: how can they be built?* Knowledge and Information Systems, 6(4):441–464, 2004.

[15] D. L. McGuinnes and F. Van Harmelen. *OWL Web Ontology Language overview*. Technical report, W3C Recommendation, 2004. Available at: http://www.w3.org/TR/2004/REC-owl-features-20040210.

[16] E. Simperl, M. Mochol, T. Bürger, and I. O. Popov. *Achieving Maturity: the State of Practice in Ontology Engineering in 2009*. International Journal of Computer Science and Applications, 7(1):45–65, 2010.

[17] M. Grüninger and M. Fox. *Methodology for the design and evaluation of ontologies*. In Proc. of the International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.

[18] M. Uschold and M. King. *Towards a methodology for building ontologies*. In Proc. of the International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.

[19] Y. Sure, S. Staab, and R. Studer. *Chapter on Ontology Engineering Methodology*, pages 135–152. Springer, Berlin-Heidelberg, 2009.

[20] M. Fernández, A. Gómez-Pérez, and N. Juristo. *METHONTOLOGY: from ontological art towards ontological engineering*. In Proc. of the American Asociation for Artificial Intelligence (AAAI) Spring Symposium Series on Ontological Engineering, pages 33–40, Stanford, USA, 1997.

[21] K. Kotis and G. A. Vouros. *Human-centered ontology engineering: The HCOME methodology*. International Journal of Knowledge and Information Systems (KAIS), 10:109–131, 2006.

[22] P. Spyns, Y. Tang, and R. Meersman. *An ontology engineering methodology for DOGMA*. Applied Ontology, 3(1–2):13–39, 2008.

[23] E. Sanders and P. J. Stappers. *Co-creation and the new landscapes of design*. Co-design, 4:5–18, 2008.

[24] D. Schuler and A. Namioka. *Participatory Design: Principles and Practices*. CRC/Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1993.

[25] C. E. Kuziemsky and F. Lau. *A four stage approach for ontology-based health information system design*. Artificial Intelligence in Medicine, 50(3):133–148, 2010.

[26] A. L. Rector, J. E. Rogers, P. E. Zanstra, and E. van der Haring. *OpenGALEN: Open Source Medical Terminology and Tools*. In Proc. of the annual American Medical Informatics Association (AMIA) Symposium, page 982, Washington, DC, USA, 2003. Available at: http://www.opengalen.org/.

[27] J. A. Blake and M. A. Harris. *The Gene Ontology (GO) project: structured vocabularies for molecular biology and their application to genome and expression analysis*. Current Protocols in Bioinformatics, 23(7.2.1–7.2.9):1472–6947, 2008. Available at: http://www.geneontology.org/.

[28] F. Ongenae, F. De Backere, K. Steurbaut, K. Colpaert, W. Kerckhove, J. Decruyenaere, and F. De Turck. *Appendix B: overview of the existing medical and natural language ontologies which can be used to support the translation process*. BMC Medical Informatics and Decision Making, 10(3):4, 2011.

[29] E. T. Miller, C. Deets, and R. Miller. *Nurse call and the work environment: lessons learned*. J Nurs Care Qual, 15(3):7–15, 1997.

[30] F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck. *An ontology-based nurse call management system (oNCS) with probabilistic priority assessment*. BMC Health Services Research, 11:28, 2011.

[31] F. Ongenae, P. Duysburgh, M. Verstraete, N. Sulmon, L. Bleumers, A. Jacobs, A. Ackaert, S. De Zutter, S. Verstichel, and F. De Turck. *User-driven design of a context-aware application: an ambient-intelligent nurse call system*. In Proc. of the User-Centered Design of Pervasive Healthcare Applications Workshop (U-CDPHA) of the 6th International Conference on Pervasive Computing Technologies for healthcare (PervasiveHealth), page 6, San Diego, CA, USA, 2012.

[32] J. Pruitt and T. Adlin. *The persona lifecycle: keeping people in mind throughout product design*. Morgan Kaufmann Publishers Inc., San Mateo, USA, 2006.

[33] S. L. Star and J. R. Griesemer. *Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–39*. Social Studies of Science, 19(3):387–420, 1989.

[34] L. Bleumers, N. Sulmon, F. Ongenae, A. Jacobs, M. Verstraete, M. Van Gils, A. Ackaert, and S. De Zutter. *Towards ontology co-creation in institutionalized care settings*. In Proc. of the 5th International Conference on Pervasive Computing Technologies for healthcare (PervasiveHealth), Dublin, Ireland, 2011.

[35] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, M. Van Gils, A. Jacobs, S. De Zutter, P. Verhoeve, A. Ackaert, and F. De Turck. *Participatory design of a continuous care ontology: towards a user-driven ontology engineering methodology*. In Proc. of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, 2011.

[36] *Boone International, the wallbed specialist*. http://www.boone-wallbeds.com/, 2013.

[37] *Televic Healthcare*. http://www.televic-healthcare.com/en/, 2013.

[38] *Dominiek Savio Institute vzw*. http://www.dominiek-savio.be/, 2013.

[39] *OLV Hospital Aalst*. http://www.olvz.be/, 2013.

[40] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers Inc., San Francisco, USA, 1997.

[41] *XMind, Professional & Powerful Mind Mapping Software*. http://www.xmind.net/, 2013.

[42] *PRoF: Patient Room of the Future*. http://www.prof-projects.com/, 2013.

[43] S. Verstichel, E. De Poorter, T. De Pauw, P. Becue, B. Volckaert, F. De Turck, I. Moerman, and P. Demeester. *Distributed ontology-based monitoring on the IBBT WiLab.t infrastructure*. In Proc. of the 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), pages 509–525, Berlin, Germany, 2010.

[44] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. *OWL-S: Semantic Markup for Web Services*. Technical report, W3C Member Submission, 2004. Available at: http://www.w3.org/Submission/OWL-S/.

[45] M. J. O'Connor and A. K. Das. *A lightweight model for representing and reasoning with temporal information in biomedical ontologies*. In Proc. of the International Conference on Health Informatics (HEALTHINF), pages 90–97, Valencia, Spain, 2010.

[46] T. H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. In Proc. of the 3rd International Semantic Web Conference, pages 229–243, Hiroshima, Japan, 2004. Available at: http://protege.stanford.edu/.

[47] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL Reasoner*. J Web Semant, 5(2):51–53, 2007. Available at: http://pellet.owldl.com/.

[48] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Technical report, W3C Member Submission, 2004. Available at: http://www.w3.org/Submission/SWRL/.

[49] V. Mascardi, V. Cordi, and P. Rosso. *A Comparison of Upper Ontologies*. Technical report, Technical Report DISI-TR-06-21, 2006. Available at: http://www.disi.unige.it/person/MascardiV/Download/DISI-TR-06-21.pdf.

[50] C. Fellbaum. *Chapter 10: WordNet*. Springer, Berlin-Heidelberg, 2010.

[51] *Cyc*. http://www.cyc.com/, 2013.

[52] *The Upper Mapping and Binding Exchange Layer (UMBEL)*. http://www.umbel.org/, 2013.

[53] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. *WonderWeb Deliverable D18: Ontology Library (The DOLCE ontology)*. Technical report, The WonderWeb Project, Laboratory for Applied Ontology, 2003. Available at: http://www.loa.istc.cnr.it/Papers/D18.pdf.

[54] *The Basic Formal Ontology (BFO)*. http://www.ifomis.org/bfo, 2013.

[55] H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, and H. Michalek. *General formal ontology (GFO): A foundational ontology integrating Object and Processes, Part I: Basic principles*. Technical report, Technical Report 8, Research Group Ontologies in Medicine (Onto-med), 2006. Available at: http://www.onto-med.de/Archiv/ontomed2002/en/theories/gfo/part1-drafts/gfo-part1-v1-0-1.pdf.

[56] I. Niles and A. Pease. *Towards a Standard Upper Ontology*. In Proc. of the International Conference on Formal Ontology in Information Systems

(FOIS), pages 2–9, New York, NY, USA, 2001. Available at: http://www.ontologyportal.org/.

[57] *The SWRL Temporal Built-In Library*. http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTemporalBuiltIns, 2013.

[58] P. Barnaghi, M. Compton, O. Corcho, R. G. Castro, J. Braybeal, A. Herzog, K. Janowicz, H. Neuhaus, A. Nikolov, and K. Page. *Semantic Sensor Network XG Final Report*. Technical report, W3C Incubator Group Report, 2011. Available at: http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/.

[59] *The DEUS Project, Design and Easy Use of wireless Services*. http://www.iminds.be/en/research/overview-projects/p/detail/deus, 2010.

[60] J. Fan, K. Barker, B. Porter, and P. Clark. *Representing Roles and Purpose*. In Proc. of the International Conference on Knowledge Capture (K-Cap), pages 38–43, Victoria, British Columbia, Canada, 2001.

[61] E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. *Organizing role-concepts in ontology development environment: Hozo*. AI Technical Report, 1(4):453–468, 2004.

[62] F. Steimann. *The role data model revisited*. Applied Ontology, 2(2):89–103, 2007.

[63] C. Bachman and M. Daya. *The role concept in data models*. In Proc. of the 3rd International Conference on VLDB, pages 464–476, 1977.

[64] N. Guarino. *Attributes and arbitrary relations*. Data and Knowledge Engineering, 8, 1992.

[65] F. Steimann. *On the representation of roles in object-oriented and conceptual modelling*. Data and Knowledge Engineering, 35(1):83–106, 2000.

[66] J. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison Wesley Publishing Company, New York, NY, USA, 1984.

[67] S. Moralee, M. Uschold, M. King, and Y. Zorgios. *The enterprise ontology*. The Knowledge Engineering Review, 13, 1998.

[68] *The Family SWRL Ontology*. http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl, 2013.

[69] A. Gangemi, S. Borgo, C. Catenacci, and J. Lehman. *Task taxonomies for knowledge content (deliverable D07)*. Technical report, Laboratory for Applied Ontology (LOA), 2005. Available at: http://www.loa.istc.cnr.it/Papers/D07_v21a.pdf.

[70] *The Process Ontology*. http://en.wikipedia.org/wiki/Process_ontology, 2013.

[71] *Semantics Utilised for Process management within and between EnteRprises (SUPER Ontologies)*. http://www.ip-super.org/content/view/129/136/, 2013.

[72] *The Process Specification Language (PSL)*. http://www.mel.nist.gov/psl/index.html, 2013.

[73] *Linked data - Connect Distributed Data across the Web*. http://linkeddata.org/, 2013.

[74] *The Business Management Ontology (BMO)*. http://www.bpiresearch.com/Resources/RE_OSSOnt/re_ossont.htm, 2013.

[75] R. Power. *Deriving rethorical relationships from semantic content*. In Proc. of the 13th European Workshop on Natural Language Generation (ENLG), pages 82–90, Nancy, France, 2011.

[76] N. Guarino and C. A. Welty. *A formal ontology of properties*. In Proc. of the 12th European Workshop on Knowledge Acquisition, Modeling and Management, pages 97–112, Juan-les-Pins, France, October 2–6 2000.

[77] J. Völker, D. Vrandečić, Y. Sure, and A. Hotho. *AEON - An approach to the automatic evaluation of ontologies*. Applied Ontology, 3(1–2), 2008.

[78] A. Miles and S. Bechhofer. *SKOS Simple Knowledge Organization System Reference* [online]. 2009. http://www.w3.org/TR/2009/REC-skos-reference-20090818/.

3

Ambient-aware Continuous Care through Semantic Context Dissemination

F. Ongenae, J. Famaey, S. Verstichel, S. De Zutter, S. Latré, A. Ackaert, P. Verhoeve and F. De Turck.

Revisions submitted to BMC Medical Informatics and Decision Making.

"Accio Brain!"

– J.K. Rowling, Harry Potter and the Order of the Phoenix (2003)

The amount of heterogeneous data provided by the monitoring equipment, captured in medical databases and generated by the available software is vast. Capturing all this data in a centralized knowledge component severely deteriorates the performance and scalability of the various context-aware applications built on this model. To tackle this issue, the O'Care Platform is proposed in the chapter. In this platform the knowledge model is distributed across the various healthcare services. These services import (a subset of) the low-level continuous care ontologies, discussed in Chapter 2, to model the needed context information and continuous care knowledge. The O'Care Platform then uses a Semantic Communication Bus (SCB) in combination with the continuous care core ontologies,

presented in Chapter 2, to adequately filter the huge amount of provided heterogeneous care data. By registering filter rules with the SCB, the healthcare services only receive the data that they are interested in at that time. To fulfill the needed scalability constraints, the SCB employs a cache and is deployed in a distributed manner. A thorough performance evaluation of the SCB in the healthcare domain was performed by an illustrative scenario concerning three healthcare applications, namely a sophisticated nurse call system supported by a localization and home automation component.
This chapter thus discusses Contribution 3 as highlighted in Section 1.3 of Chapter 1. The prototype, which is evaluated in the use case scenario of this chapter, is also the prototype, which was used in the evaluation of the participatory ontology engineering methodolody in the previous chapter. As mentioned previously, the dynamic nurse call system is discussed more thoroughly in the following chapter and Appendices B and A.

⋆ ⋆ ⋆

Abstract
Background: The ultimate ambient-intelligent care room contains numerous devices to monitor the patient, sense and adjust the environment and support the staff. The adoption of such a sensor-based approach results in a large amount of data, which can be processed by different current and future applications, e.g., task management and alerting systems. Today, the nurse is responsible for coordinating all these applications and data sets, which lowers the added value and slows down the adoption rate.

The aim of the presented research is (1) the development of a semantic model of the continuous care data that reflects the daily work practices and (2) the design of a context-aware and pervasive framework that, using this model, integrates and dynamically filters the relevant context information for the various applications.

Methods: The developed Ontology-based Care Platform (O'Care Platform) allows applications to dynamically generate and register filter rules, so that only contextual information, which is of interest at that time, is received. To semantically filter the received information according to these rules, the Semantic Communication Bus (SCB) is introduced, which uses the continuous care ontology. As the platform's proper adoption mainly depends on the correctness and completeness of the ontology, a participatory methodology was developed to co-create this model together with all the stakeholders.

Results: A prototype implementation is presented consisting of a sophisticated nurse call system supported by a localization and a home automation application component. The amount of data that is filtered and the performance of the SCB are evaluated by testing the prototype in a living lab environment. The delay introduced by processing the filter rules is negligible when 10 or fewer rules are

registered.

Conclusions: The O'Care Platform allows disseminating relevant care data for the different applications and additionally supports composing complex applications from a set of smaller independent components. This way, the platform significantly reduces the amount of information that needs to be processed by the nurses. The delay resulting from processing the filter rules is linear in the amount of rules. Distributed deployment of the SCB and using a cache allow to further improve these performance results.

3.1 Background

3.1.1 Introduction

Since a number of years, the complexity of institutional care settings has been increasing due to societal factors such as the growth of the care unit size, the more specialized nature of the care and the reduction in staffing levels. This requires a more optimized rostering and use of staff resources.

Information technology is often introduced [1] to deal with these issues. The current institutional care settings contain numerous devices to support caregivers in carrying out their everyday activities, e.g., electronic medical records and monitoring equipment. However, this high amount of technology further increases the complexity of these daily activities, because the caregivers are directly faced with complex technologies [2]. The caregiver has to use several devices to consult and insert data even when carrying out a single task. This is very time-consuming. Due to this inadequate integration of the technology, as well as the large amount of data being generated by the devices and the heavy workload of staff members, it is not rare for important events to be missed, e.g., early indications of worsening condition of a patient.

Consider for example a patient with a concussion, who needs to be in a dark environment. Today, the staff members are responsible for switching on the lights at the appropriate level each time they enter the room. Consequently, each staff member has to be aware of all the aspects and specificities of the patient's condition. If an uninformed person enters the room or a wrong button is pressed, this can cause physical pain for the patient. However, if the lighting control system would be aware of the patient's pathology and needs, it can automatically turn on the light to the correct level when it detects that the nurse enters the room. Moreover, a message can be displayed explaining to the nurse why the lights are lit on this lower level. Staff members are able to overrule the system, but a light sensor could be used to monitor the light intensity in the room and alert a nurse if a pre-defined threshold is crossed.

3.1.2 Ambient-aware continuous care

The ultimate ambient-intelligent care room of the future uses numerous devices to sense the needs and preferences of the caregivers and patients and adapt itself accordingly [3]. This implies an emerging demand for the integration and processing of the heterogeneous data offered by the different technologies available in the room.

The ACCIO [4] (Ambient-aware provisioning of Continuous Care for Intramuros Organizations) project aims to realize this goal by developing a context-aware, ambient-intelligent, pervasive and semantic platform which exploits, integrates and filters the large amount of available heterogeneous data to keep a manageable overview. This platform, called the Ontology-based Care Platform (O'Care Platform), enables technology to blend into the background, using sensors and actuators to sense and adapt to the environment according to the situation at hand [5]. This frees the caregiver from the cumbersome task of managing the different technologies. However, to achieve this goal, the platform must be able to interpret the meaning and adequately filter the relevant information out of the huge amount of heterogeneous care data provided by the sensors. Unorganized, unprocessed raw data can be voluminous, but has no meaning on itself as it has no relationships or context. Information is data that has been given meaning by defining relational connections. For this, the platform uses an ontology [6], which is a semantic model that formally describes the concepts in a certain domain, their relationships and atrributes. In this way, an ontology encourages re-use and integration. By managing the data about the current context in an ontology, intelligent algorithms can be more easily defined that take advantage of this information to automate, optimize and personalize the continuous care of patients. Referring back to the previous example, this means that the ontology models, a.o., the patient's condition and the nurse's location. Algorithms have been defined that automatically put the light to the correct level based on the context information in the ontology. Afterwards, the nurse can decide to overrule this decision by adjusting the light level in the room manually so that unexpected events can be handled.

3.1.3 Objective & paper organization

The goal of this paper is to formulate an answer to the following research questions: (1) How do we dynamically filter the large amount of data such that the different application components only receive the data that is relevant to them at that moment? (2) How do we model the continuous care data gathered and communicated between the different architecture components in a formal and semantic manner? (3) How can intelligent algorithms and applications that optimize continuous care processes be built based on this developed model?

The remainder of this article is structured as follows. The *Related work* Section highlights the contributions of this paper in view of the relevant related work. The *Methods* Section starts with a description of the architecture of the O'Care Platform in the *Architecture of the O'Care Platform* Subsection. The *Continuous care ontology* Subsection describes the developed continuous care ontology and the method which was designed to create it together with the stakeholders. The *Use case: optimizing continuous care through an ontology-based nurse call system* Subsection elaborates on the specifics of the platform using an illustrative example, namely realizing a sophisticated nurse call system supported by a Localization and Home Automation Component. To test and demonstrate the advantages and performance of the O'Care Platform, the prototype was evaluated in a living lab environment. The evaluation set-up is detailed in the final subsection of the *Methods* Section. The *Results and discussion* Section evaluates the amount of data that is filtered by the O'Care Platform and the performance and scalability of the filter rules. It also discusses the potential impact of the platform on the delivery of continuous care. The conclusions are highlighted in the *Conclusion* Section.

3.2 Related work

3.2.1 Context-aware systems

Dey and Abowd [7] refer to context as "any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves". A system may be labeled as "context-aware" if it can acquire, interpret and use context information to adapt its behavior for the current context in use [8]. A number of generic context platforms have been developed to relieve application developers from the aggregation and abstraction of context information and the derivation of high-level contexts. A complete overview and classification of the literature can be found in Hong, et al. [9], while an in-depth discussion and comparison of these platforms can be found in Baldauf, et al. [10] and Xue and Pung [11].

One of the first platforms was the Context Toolkit [12], a Java framework allowing the rapid prototyping of sensor based context-aware applications. However, the Context Toolkit does not provide a common context model to enable knowledge sharing and context reasoning. Various approaches have been proposed for modeling context information, i.e., key-value, markup scheme, graphical, object-oriented, logic-based and ontology-based models. Strang and Linnhoff-Popien [13] evaluated these context modelling approaches based on six criteria, namely distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality and applicability to existing environments. They show that ontologies fulfill most of these requirements

Context-aware systems	Context model	Contex Reasoning	Reasoning Expressivity
CoBrA	centralized	logic & rule-based	RDFS & OWL-Lite
CMF	centralized	machine learning	naive Bayesian classifier
CMSANS	centralized	logic & rule-based	RDF
SOA	centralized	logic & rule-based	RDF
SOCAM	hybrid	logic & rule-based	RDFS, OWL-Lite, Jena Rules, Prolog & hybrid
Gaia	distributed	logic & rule-based	DAML + OIL
Fook, et al.	centralized	logic & rule-based	OWL-DL
Zhang, et al	centralized	logic & rule-based	OWL-DL, Jena Rules, Prolog & hybrid
ERMHAN	2 nodes	logic & rule-based	OWL-DL & Jena Rules
O'Care Platform	distributed	logic & rule-based	OWL-DL, SWRL & Jena Rules

Table 3.1: Comparison of prevalent ontology-based generic and healthcare context-aware systems to the O'Care Platform

and are the most expressive models as these formal models allow the integration and exploitation of more specific context knowledge with high-level context information using reasoner components. The most prominent examples of context-aware systems based on ontologies are the Service-Oriented Context-Aware Middleware (SOCAM) [14], the Context Broker Architecture (CoBrA) [15], the Context Managing Framework (CMF) [16], a service-oriented middleware that integrates a Context Management Service with an Awareness and Notification Service (CMSANS) [17], the service-oriented architecture for context-aware middleware in a smart home described by Kim and Choi [18] and Gaia [19]. The properties of these systems are summarized in Table 3.1.

A distinction can be made between systems that keep the context information centralized or distributed. In a centralized system, a centralized knowledge component is used, which integrates all the context information and inferences high-level knowledge by reasoning on this shared context model. Various applications can then access this knowledge by querying this shared context model. The central context server is also able to monitor context changes and send events to the interested applications. CMF, CMSANS, CoBrA and the SOA proposed by Kim and Choi use this approach. The disadvantage of the centralized approach is that the context server forms a single point of failure and a performance bottle neck. To avoid the first problem, CoBrA offers the possibility of creating broker federations. A federation consists of multiple instances of the central knowledge component, called the context broker. These brokers then periodically exchange and synchronize contextual knowledge. An advantage of this approach is that the access to the shared context model no longer depends on the availability of one single broker. Another broker from the federation can easily be used to replace the one that becomes unavailable. However, as each broker contains all the context information, performance remains an issue.

In the distributed approach, the context information is distributed across multiple components. None of the components thus have a complete overview of the current context. GAIA supports both pull- and push-based context acquisition. The first is enabled by letting the applications specify queries for specific context information. For the latter, GAIA uses communication channels. Each channel has one or more context suppliers. The applications, called consumers, can register for context information they are interested in.

SOCAM employs a hybrid approach. Applications can directly receive context information from the different context providers. However, the context providers also supply their knowledge to a central knowledge component, called the *Context Interpreter*. This Context Interpreter maintains a shared context model and derives high-level knowledge from it. It offers this process knowledge to the different applications. The applications thus receive both low-level as well as high-level context information.

The ontology-based context-aware systems also differ in the used knowledge representation language. The most common language for describing ontologies is the Web Ontology Language (OWL) [20]. Three variants of OWL exist, with different levels of expressiveness, namely OWL-Lite, OWL-DL and OWL-Full (ordered by increasing expressiveness). OWL-Lite allows expressing a classification hierarchy and simple constraints. OWL-DL is based on Description Logics(DL) [21]. Description Logics are decidable fragments of first-order logic. Consequently, the OWL-DL variant provides the maximum expressiveness possible while retaining computational completeness, decidability and the availability of practical reasoning algorithms. As a result semantic reasoners, such as Pellet [22] or Hermit [23], can be used to infer new knowledge, i.e., logical consequences, from the information captured in OWL-Lite and OWL-DL ontologies. Rules, e.g., Semantic Web Rule Language (SWRL) [24] or Jena Rules [25], can be expressed using OWL concepts. These rule languages support a wide range of built-in operators which greatly increases expressiveness of the context model. Table 3.1

summarizes the reasoning capabilities for the aforementioned context-aware platforms.

3.2.2 Context-awareness in healthcare

Context-aware computing is a research field, which considers healthcare a relevant area of application [26]. Especially pervasive healthcare is highly suitable for using context-aware systems [27]. First, there is a large amount of available information, specific healthcare situations and related tasks, which creates a potentential for cognitive overload amongst the caregivers. Second, the patients, healthcare professionals and some equipment are fairly mobile, which requires accurate localization and adaptation of the healthcare services to the environment. Third, the financial and human resources are limited. This implies a need to cut cost while improving the quality of service to an increased number of people. Although context-awareness infrastructure inlcuding more complex devices and software will add to the total cost, the reduced number of medical errors and the ability to more effectively utilize healthcare resources should lead to reduced cost. Finally, the expectation to access, process and modify healthcare information anywhere using mobile devices is another reason to use context-awareness.

Context-aware and pervasive technologies have been applied to a number of hospital use cases [26]. The following notable prototypes have been proposed in literature. The “hospital of the future” [28] prototype consists of a context-aware Electronic Patient Record (EPR) filtering information according to the current context, an intelligent pill container for proper dose administration and a context-aware hospital bed of which the content of the display changes according to the context and warns for some incorrect actions. The Context-aware Mobile-Ward [29] is designed to support nurses in conducting morning procedures in a hospital ward. An intelligent hospital prototype [30] has been developed, which allows localization of a team member and the ability to initiate an audio-video conference from the nearest point. Similarly, the Vocera communication system [31] supports communication amongst hospital workers via mobile devices and localization techniques. A context-aware mobile communication prototype [32] that empowers mobile devices to recognize the context in which hospital workers perform their tasks in order to provide contextual messaging.

Similarly several context-aware prototypes have also been developed for homecare and residential care. Prototypes to assist patients in taking their medications at home [33, 34]. Vivago [35], a social alarm system for elderly based on wearable sensors and providing long-term monitoring of user’s activity profile and automatic alarm notification. A context-aware assistant during hand washing for adults with dementia [36]. The use of context-aware systems for telemedicine of chronic diseases [37, 38]. The LifeMinder prototype [39] can sense pulse waves, user's actions and postures and capture contextual photos and continuous voices. This information is then used to detect stressful states. A prototype [40] to detect falls of elderly by using a visual fall detection system and combining this with context information, e.g., patient's general condition, location and duration of patient's inactivity. The U-Health Smart Home at POSTECH [41] to help the elderly to continue to live a more independent life as long as possible in their own home while being monitored and assisted in an unobtrusive manner.

Examples can also be found in literature of context-aware healthcare systems, based on ontologies. Fook, et al. [42] presents a context-aware system for monitoring and handling agitation behavior in persons with dementia. Zhang, et al. [43] propose a context-aware infrastructure to support the global healthcare system in terms of device access, context management and service interoperability. These two approaches adopt a centralized knowledge management approach with as central component an ontology-based knowledge component. The Emilia Romagna Mobile Health Assistance Network (ERMHAN) [44] is a multichannel context-aware service platform designed to enable the development and delivery of an extensible set of care services which allow patients to be assisted at home and support the activity and mutual collaboration of care providers who are involved in patient care and assistance. This framework distributed the context knowledge across two nodes. The *Patient Context Manager* is deployed at the patient site and is responsible for preprocessing the data retrieved by biomedical and environmental sensor networks. Rule-based reasoning is employed to detect abnormal phenomena in this sensor data and forward these alarms to the *Central Context Manager*. This *Central Context Manager* is deployed in the care centre and combines the alarms received from the *Patient Context Manager* with other context information in order to take appropriate actions. The properties of these systems are also summarized in Table 3.1.

The healthcare scenario has some specific implications which differentiate it from other scenarios. Although much research has been done on the subject, the adoption of context-aware services is lagging behind what could be expected. Most of the mentioned projects are prototypes and real applications are still difficult to find. Whereas the healthcare industry is quick to exploit the latest medical technology, they are reluctant adopters of modern health information systems [45]. Half of all computer-based information systems fail due to user resistance and staff interference [46]. The main complaint made against mobile, context-aware systems is that users had to significantly alter workflow patterns to accommodate the system [47]. An often overlooked fact is that the strength any context-aware platform is heavily dependent on the correctness and completeness of the used knowledge model. This model needs to capture the daily work practices and context of the caregivers and patients accurately [48]. Constructing this model is a difficult task. In contrast to the healthcare domain in general, a lot of knowledge used in continuous care, e.g., how to prioritize and assess nurse calls or assign caregivers to patients, is implicit and best practices are not rigorously documented.

Another challenge in the healthcare scenario is the fact that wrong decisions made by the system can have severe implications. Context data delivered by sensors is very unreliable. Decisions made based on wrong or incomplete sensor data might thus not be correct. Quality of Context-aware (QoC-aware) algorithms that take the reliability and correctness of the context data into account should thus be developed to mediate this issue.

3.2.3 Publish/subscribe systems

Publish/subscribe systems have evolved from static topic-based to dynamic content-based systems. By augmenting the content with semantics, subscriptions can be created which take into account the actual meaning of the content. Several semantic publish/subscribe systems have been proposed in literature [49] which differ in the method proposed to relate subscriptions to messages, namely based on Resource Description Framework (RDF) [50] graph-matching, ontological inferencing and attribute-value pair matching. Our approach is most closely related to semantic publish/subscribe systems that use OWL inferencing. These systems represent subscriptions as OWL concepts and messages as concept instances. An inferencing engine is used to determine if a message instance satisfies the constraints of a subscription class. This approach is more expressive than the custom RDF graph-matching algorithms as it allows new, non-asserted knowledge to be inferred. Moreover, it does not limit the format that messages are allowed to take as is the case in systems based on attribute-value pairs.

3.2.4 Ontologies for representing context in healthcare environments

The definition and use of ontologies in the medical domain is an active research field, as it has been recognized that ontology-based systems can be used to improve the management of complex health systems [51]. However, most of the developed ontologies focus on biomedical research and are mainly employed to clearly define medical terminology [52], e.g., Galen Common Reference Model [53] or the Gene Ontology [54]. Little work has been done on developing high-level ontologies, which can be used to model context information and knowledge utilized across the various continuous care settings. However, ontologies have been developed for specific subdomains of continuous care, e.g., ontologies for structuring organization knowledge in homecare assistance [51], representing the context of the activity in which the user is engaged [55] and modeling chronic disease management in homecare settings [44]. The used contextual information is often very simple. Time, location and profile information of staff members and patients are the most used contextual parameters [26]. A major challenge in modeling context-aware healthcare ontologies is that the description of a situation by using what (activity), who (identity), where (location) and when (time) may not be enough [27]. More richness and higher reliability are required. This could include how (process), with whom (sources), and so what (needed action). Moreover, the current continuous care ontologies are not co-created together with the various healthcare stakeholders. Kuziemsky and Lau [56] see ontology co-creation as the key to the challenge of creating an ontology that is both accurate and useful. They observe how little research has been done in a field that nevertheless greatly benefits from high-quality and practical ontologies: (health)care.

3.2.5 Our contribution

In this research, the O'Care Platform, a distributed, scalable, context-aware and pervasive platform to support continuous care processes, is presented. This platform employs a Semantic Communication Bus (SCB) to accomplish a flexible and semantic publish/subscribe mechanism to communicate context information between the devices delivering context information and the applications processing this information. Table 3.1 compares the O'Care Platform to the prevalent generic and healthcare ontology-based context-aware systems. It can be noted that our approach adopts a distributed context model. The SCB, which uses a set of core ontologies to model the communicated context information, forwards the gathered context information to the different applications, but does not retain this information. The applications have their own individual knowledge component, which contains a domain-specific extension of (a subset of) the core ontologies. The applications do retain the context information they obtain. To communicate to the SCB in which information they are interested, the applications register filter rules with the SCB. They are also able to post inferred knowledge back on the SCB. The SCB thus loosely couples the context providers and the applications. None of the components have a complete overview of the current context as the knowledge is distributed across the various applications. In contrast to a shared context model, the application components only manage the context model and context information pertaining to their specific domain. This improves the scalability and performance of these applications, as they need to manage less context information with more concise context models. Consequently, expressive inferencing, i.e., OWL-DL, SWRL and Jena Rules inferencing, can be efficiently performed. It can be derived from Table 3.1 that the use of a distributed context model has not found a lot of uptake in healthcare context-aware systems. Most systems use a centralized context model. ERMHAN only distributes the context model across two nodes, one at the home of the patient and one in the care center. This distribution is thus location-based. Our approach distributes the context according to application domains.

Additionally, our approach differs from other OWL inferencing publish/subscribe systems as it also allows the use of Jena rules and SWRL [24] to define subscriptions. These rule languages support a wide range of built-in operators which greatly increases expressiveness. Jena [25] rule inferencing exhibits the best scaling behavior in function of the amount of subscription rules and increasing message complexity [57, 58], but is less expressive than SWRL or OWL inferencing. The choice between these three reasoning approaches allows balancing expressivity and performance according to the specific use case at hand. The generic context-aware platform with semantic publish/subscribe mechanism presented in this paper has already been applied to several autonomic network management scenarios such as the management of a multimedia access network and the management of a cloud data center [58]. This paper thoroughly evaluates the performance of the proposed platform within the healthcare domain. Moreover, it is shown how a more scalable platform can be achieved by distributing the SCB and employing a cache.

As mentioned previously, the healthcare scenario differentiates itself from other scenarios to which the platform has been applied by the slow adoption of context-aware systems by the users and the need for QoC-aware algorithms to mediate the unreliability of the sensors. To resolve the first issue, this research proposes a participatory ontology engineering methodology which promotes user participation, while not overloading the involved stakeholders as time is a valuable resource. The methodology actively involves social scientists, ontology engineers and domain stakeholders, i.e., in this case nurses, caregivers, patients, doctors and professionals working for the healthcare industry, in the ontology engineering process. The stakeholders participate in each step of the life cycle of the ontology without having to construct the ontology themselves or attribute a large amount of their time. This ensures that the developed ontology model accurately reflects the work environment and context of the stakeholders and takes away their fear of the technology. To support the development of QoC-aware algorithms that take the reliability and correctness of the context data into account, the developed platform and accompanying ontologies provide techniques to detect unreliable sensor observations and annotate them [59]. Additionally, the users are always able to overrule the decisions of the platform. This also improves the acceptance rate of the new technology as the users feel much more in control of the system.

As mentioned previously, little work has been done on the development of high-level continuous care ontologies, which can be re-used across the various continuous care domains, e.g., hospitals, care residences and homecare. Therefore, the developed participatory ontology engineering methodology was used to design seven continuous care core ontologies and a plethora of domain-specific ontologies, which import (a subset of) the core ontologies. Including the domain expert in the creation of the ontology facilitates the acceptance of the new technology that is built using this knowledge model, since such a user-driven approach allows the domain experts to have control over the knowledge flow in their environment and adapt it to their needs. Moreover, the ontology was developed in such a way that it can easily be extended with new knowledge.

As a last contribution, this paper also presents a framework and algorithms allowing applications to autonomously generate and register new filter rules based on the current context.

In summary the contributions of this paper are:

- The design of a continuous care ontology that accurately reflects the daily work practices of the caregivers.
- The design of a participatory ontology engineering methodology providing guidelines on how an ontology can be co-created together with the stakeholders.
- The design of the O'Care Platform, which combines expressive OWL-DL context reasoning, distributed management of the context model & information, intelligent filtering of context information and distributed deployment of the publish/subscribe mechanism and employment of a cache to increase scalability. The combination of all these features differentiates this platform from other works in the same area.
- The design of a framework and algorithm allowing the application components to automatically generate new filter rules based on the received context information.
- The performance evaluation of the SCB in the healthcare domain.

3.3 Methods

3.3.1 Architecture of the O'Care Platform

The ambient-intelligent care room consists of various devices, e.g., sensors and nurse call buttons, and intelligent applications that process the generated data. A communication substrate is needed to glue these components together and orchestrate collaboration. For this, the *Semantic Communication Bus (SCB)* [58] was designed, as visualized in Figure 3.1. The SCB orchestrates the communication of semantically enriched data. This allows filtering data based on meaning rather than on string patterns. The SCB interprets the data by using *core ontologies* which model the information being exchanged for a continuous care domain. For example, the ontologies model that the environment contains light sensors, which make

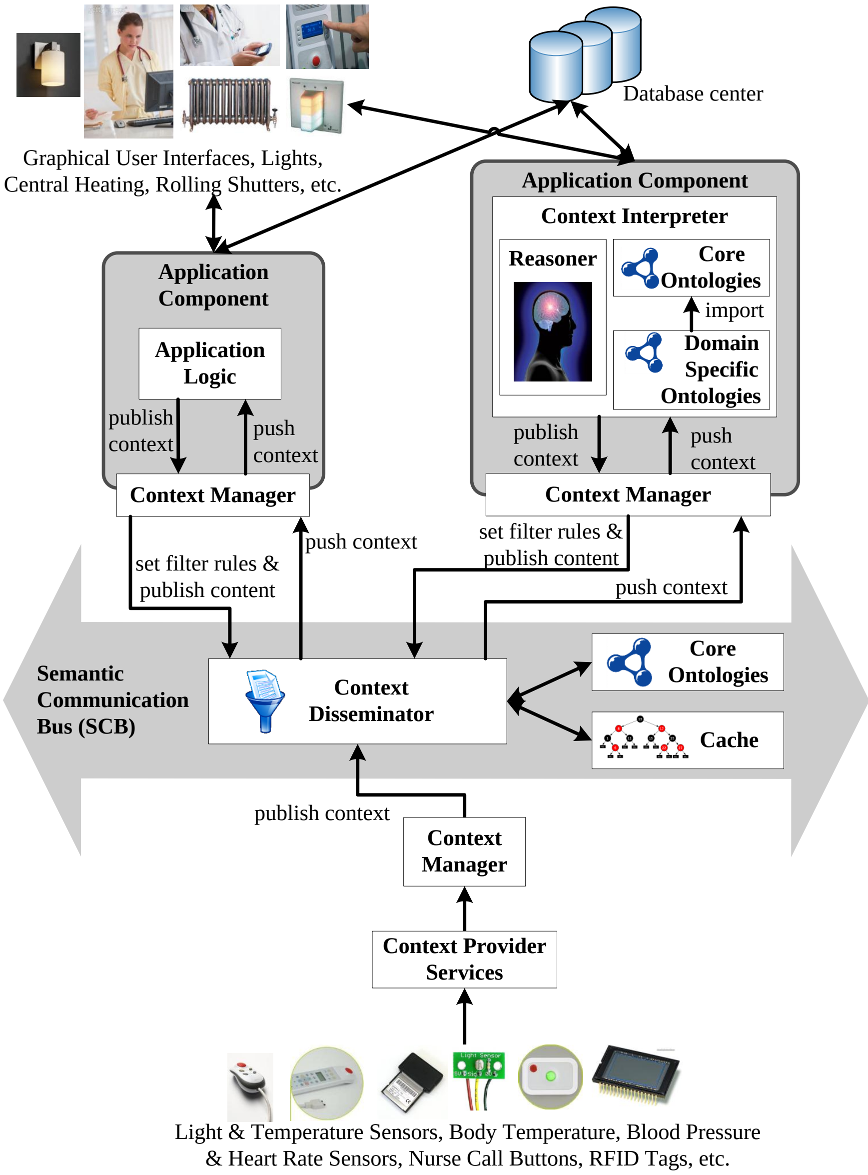


Figure 3.1: ***Architecture of the O'Care Platform using a Semantic Communication Bus for interaction, collaboration and orchestration.***

observations about the light intensity at a location. As mentioned previously, little work has been done on developing ontologies to support the continuous care of patients. Such an ontology has to contain information about the profile of staff members and patients, roles and responsibilities, administrative information, etc. To tackle this issue, a participatory ontology engineering methodology was developed to co-create the continuous care core ontologies with the stakeholders. This is further detailed in the *Continuous care ontology* Section.

As depicted at the bottom of Figure 3.1, the *Context Provider Services* receive data from the devices in the environment and transform it to context information which adheres to concepts in the core ontologies. This semantically enriched data is forwarded to the *Context Manager*, which publishes it on the SCB. For example, the *Location Provider Service* is used to publish location information, e.g., about staff members or patients, on the SCB.

As is the case with classic publish/subscribe mechanisms, the SCB allows application components to subscribe to context information, which is relevant to them at that moment, through the *Context Disseminator*. The application components use a Context Manager, which contains (a subset of) the core ontologies used by the SCB, to specify the context they are interested in, by defining *filtering rules* and registering them with the Context Disseminator. For example, a nurse alerting application component indicates that it is only interested in light intensity observations, crossing a particular threshold and coming from rooms with patients who suffer from a concussion. These rules are expressed using concepts from the core ontologies. Examples of such rules can be found in Figure 3.7 and in the Flexible and semantic publish/subscribe mechanism Section.

To ensure that the SCB can process all the events it receives in a timely manner, the Context Disseminator uses a cache [60]. When continuous care information is published on the SCB, the Context Dessiminator first checks the cache. If the published event is not found in the cache, a miss occured and the Context Disseminator matches this published event with the registered filter rules by reasoning on the information in the ontology. If a match is found, the information is forwarded to the application components that subscribed to the filter rule. This match is also added to the cache. The cache thus contains mappings from published events onto the filter rules with which these events matched. However, if the published event is found in the cache, a hit occured. The filter rules to which this event is mapped are collected and the event is forwarded to the application components that subscribed to the filter rule. Consequently, no reasoning needs to be performed to process the event. Efficient search algorithms exist to implement a cache, which allow performing a cache look-up in a very performant manner [61].

It can be noted that the use of these filter rules reduces the amount of care data that is forwarded to the application components. This prevents these components from being flooded with huge amounts of data generated by the sensors and devices in a truly pervasive and ambient-aware patient room. Moreover, application components can register new filter rules on the fly based on the current context, which greatly improves the flexibility of the platform. This is further detailed in the *Flexible and semantic publish/subscribe mechanism* Section.

As visualized in the top right of Figure 3.1, the intelligent application components, receiving context information from the SCB, also use ontologies to model their specific (sub)domain and perform sophisticated reasoning. These *domain-specific ontologies* extend concepts in the core ontologies, such that the context delivered by the SCB is directly understood by the application logic. Static information about the environment, e.g., names of patients or locations of sensors, is collected from databases. As a result of the reasoning, the application components can adapt the environment by controlling devices, e.g., lights or beepers. The application components can also publish their conclusions on the SCB through the Context Manager. This way, they can be picked up by other application components to perform additional reasoning. For example, a first application component computes the locations of people, based on the available raw sensor information, and publishes these locations on the SCB. A second application component uses this augmented location information to assign staff members to tasks, while a third application component uses it to regulate the light level in a room.

3.3.2 Continuous care ontology

The incorporation of ontology engineering tasks in knowledge-empowered organizations, such as hospitals, can prove to be a hindrance if not done in a way that is seamless to the day-to-day activities of the nurses, patients and doctors [62]. To resolve this issue, the construction of the ontology should be user-driven. This will not only facilitate the acceptance of this new technology, but it will also empower the staff to make suggestions for changes and thus shape the common information space to their needs.

The existing ontology engineering methodologies are rather extreme in their choices to include domain experts [63]. There are methodologies, e.g., Tove [64], Enterprise [65] and Methontology [66], that only discuss the scope and the requirements of the ontology with the domain experts [67]. The rest of the ontology lifecycle is controlled by the knowledge engineer. A Human-Centered Ontology engineering Methodology (HCOME) [62] was proposed, which offers user-friendly and collaborative tools that allow domain experts to construct, merge and discuss their own ontologies. The knowledge engineer only delivers (technical) support in this process.

A methodology was developed that holds a middle ground between these two extremes, so that it can be used to develop ontologies for less IT-focused domains where the stakeholders might not be willing or able to construct the ontologies themselves. The ontology engineering process actively involves social scientists, ontology engineers and stakeholders by employing several participatory methods and techniques to capture the daily and preferred practices, e.g., observations, role-playing and discussing scenarios in hands-on workshops. The designed methodology promotes user participation while taking into account that time is a valuable resource. A detailed discussion and evaluation of the methodology can be found in Ongenae, et al. [68].

The development of the continuous care ontology started with extensive ethnographic research, after which an interdisciplinary group of domain experts closely interacted with ontology engineers and social scientists in role-playing workshops to construct a conceptual model of the ontology under development. This model was then mapped on existing ontologies to to identify re-usable ones. In order to transform the conceptual model into a formal one, decision-tree workshops were organized to capture the axioms that restrict the possible interpretations of the concepts in the ontology. Once a preliminary ontology was developed, this model was implemented in a knowledge representation language and a prototype application was built using this ontology. More information about this prototype can be found in the following section. Additional role-playing workshops were then organized involving a broader group of domain experts. These workshops aimed to give the participants a first-hand experience of the developed prototype in order to generate deeper reflection on both the application and the ontology so that the applicability of the ontology across other continuous care settings could be evaluated.

Knowledge about a certain domain constantly changes such as the discovery of new drugs and diseases. An ontology is therefore not a static model, but a dynamic one that constantly should be able to change and evolve. This was taken into account by developing the continuous care ontology in such a way that the model can easily be extended with new concepts, relations and axioms.

As a first measure, a distinction was made between general and domain-specific continuous care knowledge. The first is of interest to various context-aware healthcare applications and is applicable across all continuous care domains, e.g., hospitals, homecare environments and residential care settings. This knowledge is modeled in the in the continuous care core ontologies. Adding too many axioms to the core ontologies that constrain the possible interpretations of concepts was especially avoided, unless there was very wide agreement about the constraint amongst the stakeholders involved in the co-creation process. This facilitates cross-domain applicability of the core ontologies and allows easy extension without contradicting with the knowledge already contained in these ontologies. The domain-specific ontologies, which were also developed using the participatory ontology engineering methodology, model knowledge particular to a specific domain, e.g., the specific roles and competences employed within a hospital setting and how they map on each other, the specific patient profiles present within a care residence or the particular continuous care workflows pertaining to nurse calls or care requests. All concepts in the domain-specific ontologies are always subclasses of concepts in the core ontologies. New domain-specific ontologies can thus easily be defined by extending the core ontologies.

Second, the continuous care core ontology was developed in a modular way instead of as one big semantic model. The application of the participatory ontology engineering methodology resulted in seven core ontologies for the continuous care domain, namely the *Upper*, *Sensor & Observation*, *Context*, *Profile*, *Role & Competence*, *Medical* and *Task* ontologies. These modules are linked to each other by using the OWL import mechanism. This way, the domain-specific ontologies can easily import and extend a specific core module instead of importing the whole continuous care core ontology. This facilitates re-use and allows application components to only use a subset of the continuous care core ontology to perform the domain-specific reasoning. A smaller, focused ontology is also easier to interpret and extend with new concepts, relations and definitions.

As a final measure, the defined concepts are grouped as much as possible into logical categories, according to the properties that they share. The logical category is introduced as a concept in the ontology and the grouped concepts are defined as subclasses of this concept. For example, consider the process of modeling the profile information of a person, e.g., his/her mother tongue, sex and nationality. This information could be represented by relationships with as domain the `Person` concept, e.g., `hasSex` or `hasNationality`, or as separate concepts in the ontology, e.g., `Sex` or `Nationality`. In the continuous care ontology the latter approach is chosen and these concepts are introduced as subclasses of a logical category concept, i.e., `Profile`. Restrictions, relations and properties can then be defined on this logical category. When new profile information needs to be added to the ontology, it can easily be added as a subclass of the logical category concept. The new profile information will automatically inherit all the relations, restrictions and properties defined on the `Profile` concept. This makes it easier to manage and extend the ontology. As can be seen at the bottom of Figure 3.2, the logical category can even be further divided in several logical subcategories, e.g., the division of the `Profile` concept into the `Biological`, `Psychological` and `Sociological Profile` concepts.

The continuous care core ontologies are used by the SCB to filter the context information for the appropriate application components. The most important concepts and relations of the core ontologies, with respect to the use case detailed in the following section, are depicted in Figure 3.2 and discussed in the following paragraphs. The application components use (a subset of) the core and domain-specific ontologies to perform sophisticated reasoning on the data they receive from the SCB.

The *Upper ontology* describes general classes, relations and axioms. Most importantly this ontology enables data to be related with a unique ID. The classes preceded by the namespace prefix `temporal` are imported from the *SWRL Temporal Ontology* [69] and model complex interval-based temporal information. All

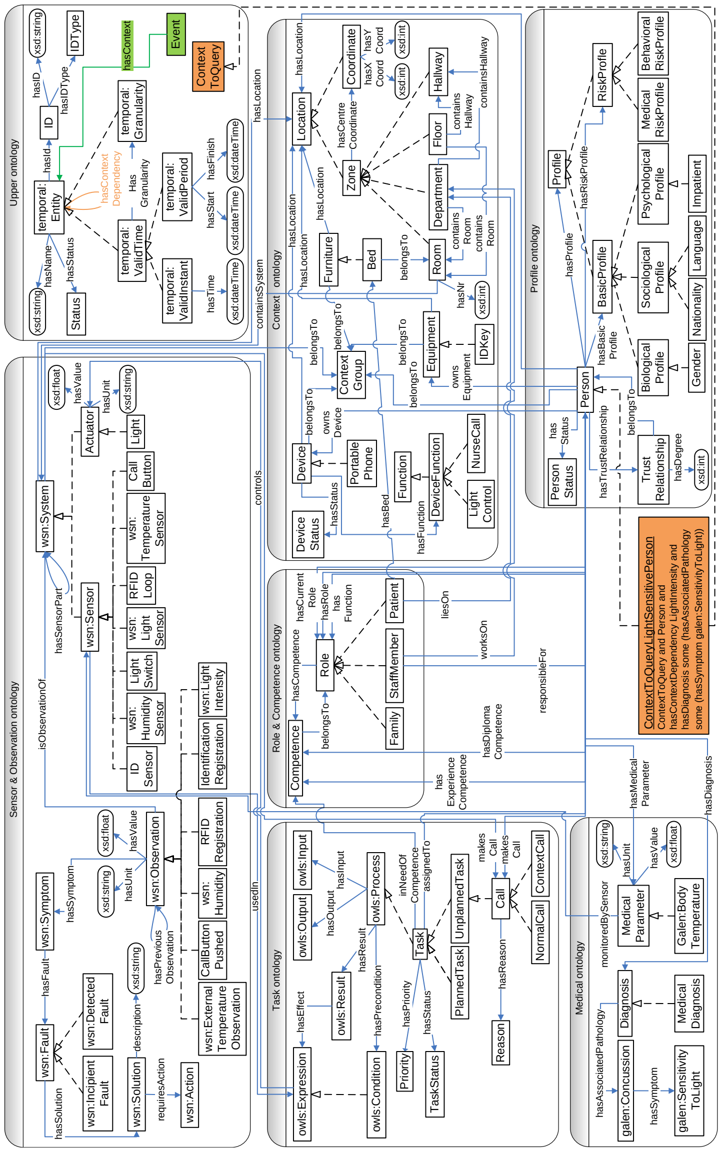


Figure 3.2: ***Overview of the most prevalent classes and relations of the seven continuous care core ontologies.*** *The figure visualizes the most prevalent classes, their relations and properties of the seven continuous care care ontologies, namely the Upper, Sensor, Context, Profile, Role & Competence, Medical and Task ontology. The squares represent the classes. The dashed arrows depict subclass relationships. The blue arrows represent relationships between classes (object properties) and between classes and ranges (datatype properties). Ovals represent the ranges of the datatype properties.*

the other core ontologies import this ontology and define all their concepts as subconcepts of `temporal:Entity`. This is not shown on Figure 3.2 to avoid overloading the figure.

The *Sensor & Observation ontology* is one of the most important ontologies for filtering data. The concepts preceded by the `wsn` namespace prefix are imported from the *Wireless Sensor Network ontology* (WSN) [59], which was developed by the co-authors and allows giving meaning to data values monitored by sensors. The `System` concept models a system and its components. An `Observation` represents a data value monitored by a system. However, context information is often unreliable as it is gathered by sensors which can be imprecise or erroneous, e.g., fall detection sensors are known to often generate false positives. Moreover, the context information can be ambiguous as information gathered by different sensors can be conflicting or the context information might even be incomplete if there is no sensor information available. As it is this context information that determines the behavior and the strategies of the different application components, it is important to make the quality of the context data explicit to prevent error propagation. To support the development of QoC-aware algorithms, this ontology contains axioms and rules, modeled as `Symptom` concepts, which allow detecting specific phenomena in the observations published to the SCB. For example, the `LightIntensityBelowZeroSymptom` detects light intensity observations that are below zero. Using OWL2 DL mechanisms, axioms are provided that reclassify these symptom individuals as `Fault` concepts, e.g., the previously mentioned symptom is reclassified as a `FaultyLightIntensitySensor` indicating that the sensor that made the measurement is faulty, since light intensity can never be below zero. Additionally, a fault can be reclassified as a `Solution`, e.g., the previous fault is reclassified as the `DoNotUseSensor` solution indicating that measurements from this sensor should not be used by the algorithms. Consequently, the application components can take these classifications into account in their filter rules and algorithms. For example, on the one hand, the application components can register filter rules that indicate that observations annotated as `FaultyLightIntensitySensor` concepts should not be forwarded to the application component. On the other hand, application components could also choose to receive these annotated observations and process them differently in their own algorithms, e.g., for fault detection and diagnosis. The WSN ontology was extended with system, sensor, actuator concepts and their associated observation, fault and solution concepts that play an important role in continuous care settings, e.g., nurse call buttons.

The *Context ontology* models the contextual environment information. `ContextGroup` is the most important concept. It represents a logical grouping of entities that belong together, e.g., a patient with all his/her devices, room, bed and other equipment. The composition of a context group dynamically changes based on the context. This ontology also contains all the information related to localization. A `Location` either can be a coordinate or a zone.

The *Profile Ontology* models the profile information about staff members and patients that was indicated as being important by the stakeholders in the workshops. Each `Person` is associated with a `Profile`, which consists of a basic and a risk profile. The latter is defined by axioms and rules, which allows inferencing the risk profile of the patient by reasoning on the information in the basic profile.

The *Role & Competence ontology* defines each role by its competences through axioms. This supports algorithms that find the most appropriate staff members to fulfill a task based on the required competences. Each person is then associated with competences and roles through five relationships: `hasFunction`, `hasRole`, `hasCurrentRole`, `hasDiplomaCompetence` and `hasExperienceCompetence`. The `hasFunction` relation models the primary role of a person, i.e., the role for which this person was primarily hired. The `hasRole` relation indicates all the roles a person can fulfill, while the `hasCurrentRole` models the person's current role. If the latter is not instantiated, it is assumed that the current role of the person is his/her function. The `hasDiplomaCompetence` and `hasExperienceCompetence` indicate extra competences a person has acquired by either following courses or through experience.

The *Galen Common Reference Model* [53], of which the concepts are preceded by the `galen` namespace prefix, represents clinical terminology. The *Medical ontology* adds axioms and constraints to this imported terminology, which express relations between this medical knowledge and concepts in the other ontologies.

Finally, the *Task ontology* models continuous care process workflows. A workflow represents a sequence of related continuous care tasks, which are conducted in a particular order. For this, the *OWL-S Process ontology* [70] was imported, of which the classes are preceded by the `owls` namespace prefix. The `Process` concept models a process, which can return information and produce a change in environment based on the context and the information it is given. This is described by `hasInput`, `hasOutput`, `hasPrecondition` and `hasEffect` relations. A process can be composed of several other processes. The `Task` concept, introduced as subclass of `Process`, represents the various continuous care tasks. Consider, for example, the task of assigning a person to a call. A `Call` is modeled as an unplanned task. A `Normal Call` is then modeled as a `Call`, which has as precondition that a patient pushes a call button. This task takes the patient as input and the assigned caregivers as output. The effect of the `Normal Call` is that the assigned caregivers' cellphones ring.

3.3.3 Use case: optimizing continuous care through an ontology-based nurse call system

3.3.3.1 Scenario description

Nurse call systems are a very important fundamental technology in continuous care as they are used by caregivers to coordinate work, be alerted of patients' needs, communicate with them through intercoms and request help from other staff members. When patients feel unwell they push a button. The nurses then receive a message in a beeper with the room number. This brings the question: which nurse goes to the room, the closest one, the one on call, etc.? The current systems have oftentimes a very static nature as call buttons have fixed locations, e.g., on the wall next to the bed. There is an increased risk when patients become unwell inside a hallway, staircase or outside as they cannot use the nurse call system. Additionally, the current nurse call algorithms consist of predefined links of beeper numbers to rooms. Consequently, the system presently does not take into account the various factors specific to a given situation, such as the pathology of a patient, e.g., heart patient or confused, nor the competences of the staff, e.g., nurse or caregiver.

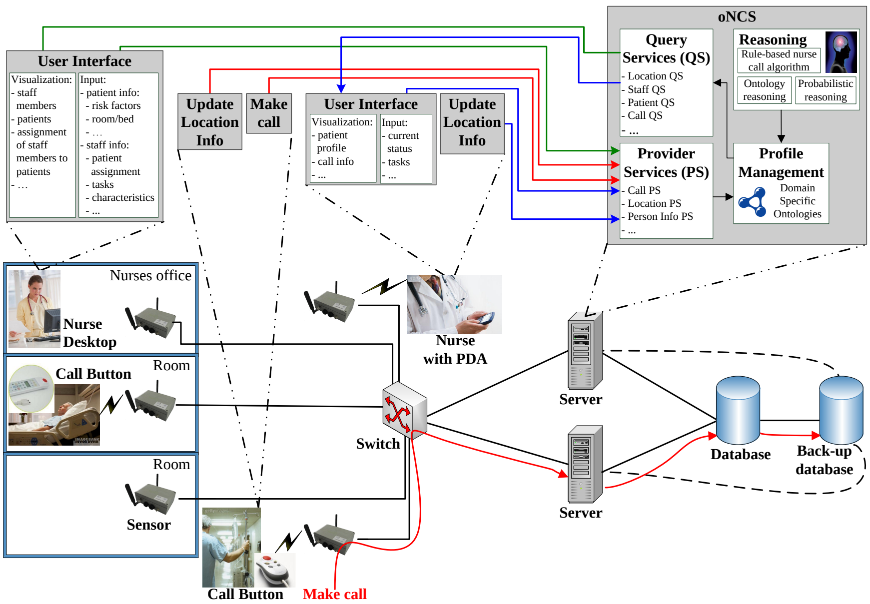


Figure 3.3: ***General concept of the oNCS System with probabilistic priority assessment and profile management.***

The increased introduction of electronic devices in continuous care settings facilitated the development of the ontology-based Nurse Call System (oNCS), visualized in Figure 3.3, which allows patients to walk around freely and use wireless nurse call buttons. Additionally, this platform manages the profiles of staff members and patients in an ontology. A sophisticated nurse call algorithm was developed by the authors. It first determines the priority of the call using probabilistic reasoning algorithms, which take into account the origin of the call and the pathology of the patient. Next, the algorithm finds the most appropriate staff member to handle the call. It dynamically adapts to the situation at hand by taking into account the context, e.g., location of the staff members and patients. A detailed description of this platform can be found in Ongenae, et al. [71].

To better illustrate the benefits of the O'Care Platform, an extension of the oNCS Application Component is presented in this paper, which allows nurse calls to be automatically launched based on the data generated by the electronic equipment and sensors in the environment, e.g., alerting a nurse when the light intensity is too high in the room of a patient with a concussion. The proposed extension provides a solution to potential risky situations being missed because the caregivers are overloaded with constantly monitoring and orchestrating all the devices in the ambient patient room, making it more applicable to real-time scenarios.

To realize this extension, two other application components were designed, namely a Localization and Home Automation Application Component. The first determines the location of patients, staff members and devices. The latter automatically controls the ambient-intelligent activity in the room of the patient, e.g., switching on the lights at the appropriate level. The oNCS, the Home Automation and Localization Application Component each represent an *Application Component* which uses the SCB to filter the context information that is relevant for it at that moment. The architecture of these components is thus very similar to the architecture of the *Application Component* at the top right of Figure 3.1. These three application components, the SCB, the sensors and the actuators together form a scenario-specific implementation of the O'Care Platform.

3.3.3.2 Flexible and semantic publish/subscribe mechanism

The SCB allows application components to subscribe to relevant context information by registering filter rules with the Context Disseminator. When a context publisher forwards information to the SCB, the Context Disseminator reasons on the core ontologies to determine which subscribers are interested in this information by computing whether the forwarded data satisfies at least one filter rule defined by the subscriber. If it does, the information is forwarded to the application component.

Publishing context: The Context Provider Services are used to semantically annotate data with concepts from the core ontologies so that it can be interpreted by the SCB. To achieve a flexible system, in which published context and filter rules can easily be matched, an `Event` concept is added to the *Upper Ontology*, which has a `hasContext` relationship to `temporal:Entity` concepts. These additions are indicated in green in Figure 3.2. As all the other core ontologies import the *Upper Ontology*, this `Event` concept and `hasContext` relationship essentially becomes a part of all the other core ontologies too. These core ontologies have defined all their concepts as subconcepts of `temporal:Entity`. As such, all their instances can occur as range of the `hasContext` relation. As such, new types of events and filter rules can easily be expressed and matched. No other modifications to the ontologies are needed to support the management of these events.

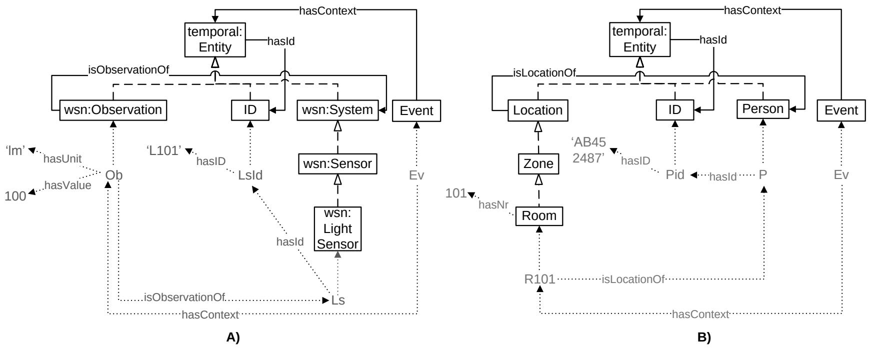


*Figure 3.4: **Examples of events published on the SCB.** This instantiation diagram visualizes two examples of events that are published on the SCB, namely a) an event indicating that a light sensor with ID L101 measured a light intensity of 100 lumen and b) an event expressing that a person with ID AB452487 is located in room 101. The squares represent classes from the core ontologies. The dashed arrows depict subclass relationships. The black arrows represent relationships between classes (object properties). The gray text and dotted arrows indicate instances of classes, ranges of datatype properties and instantiated object properties.*

When a device wants to publish context information, the Context Provider Service creates an event, containing the data it wishes to publish. For example, to publish that the light sensor with ID L101 measured a light intensity of 100 lumen, the following instances are created (see also Figure 3.4a):

- Ob **owl:instanceOf** Observation **and** Ob hasValue 100 **and** Ob hasUnit 'lm'
- LsId **owl:instanceOf** ID **and** LsId hasID 'L101'
- Ls **owl:instanceOf** LightSensor **and** Ls hasId LsId **and** Ob isObservationOf Ls
- Ev **owl:instanceOf** Event **and** Ev hasContext Ob

Similarly, application components can publish the results of their reasoning. For example, the Localization Component collects all the context information that gives an indication of the location of staff members, such as Radio-Frequency IDentification (RFID) tags, and calculates their current locations, e.g., room 101. These locations are published on the SCB to be used by other application components, e.g., the oNCS Application Component, as follows (see also Figure 3.4b):

- Pid **owl:instanceOf** ID **and** Pid hasID 'AB452487'
- P **owl:instanceOf** Person **and** P hasId Pid
- R101 **owl:instanceOf** Room **and** R101 hasNr 101 **and** R101 isLocationOf P
- Ev **owl:instanceOf** Event **and** Ev hasContext R101

To support the aggregation of context data and allow filter rules to process multiple observations simultaneously, more complex Context Provider Services can be written. For example, instead of publishing each light intensity observation to the SCB separately as in the first example, a Context Provider Service was developed to calculate the average of all the light intensity observations in a room within a minute and only publish this average as an event on the SCB. This event can be published as a normal `LightIntensity` observation or the aggregation process can be made explicit to the SCB by creating a new type of the `Observation` concept in the *Sensor & Observation core ontology*, for example `Averaged-LightIntensityObservation`. Filter rules can then be written to process this new type of events. Similarly, Context Provider Services can be written to aggregate the values of different types of sensors.

Subscribing to context: Filter rules are expressed by defining subclasses of the `Event` class. This way, the Context Disseminator can determine if the published context matches a filter rule by asking an OWL Reasoner, such as Pellet, if the published event can be classified as an instance of the `Event` defined by the filter rule [58]. The filter rule is defined by means of necessary and sufficient conditions, which the published event must fulfill to belong to this class. Moreover, the Context Disseminator can use the OWL reasoner to check if the filter rule is satisfiable, i.e., does not contradict with the knowledge already defined in the core ontologies. If the filter rule is unsatisfiable, the subscription of the filter rule fails and the class is not added to the ontology. This ensures increased robustness.

For example, as the Home Automation Component regulates the ambient-intelligent activity in the room of a patient, it is interested in location information as well as light intensity, humidity and temperature observations. It registers the following filter rule:

Event **and** hasContext **some** (LightIntensity
or ExternalTemperatureObservation
or Humidity
or Location)

Note that the two `Event` examples match with this filter rule and will thus be forwarded to the Home Automation Component. The first example matches because the ontology declares that each `Observation`, which has a unit of type lumen, is an observation of type `LightIntensity`. Similarly, the second event matches because the ontology states that each `Room` is a subclass of `Location`.

Generating new filter rules: The information in which an application component is interested can change based on the current context. Instead of filtering all the context that might be needed at some point in time, the application components are able to generate new filter rules when the context changes. The filter rule generation process is thus made context dependent. To enable this, the ontology allows defining context dependencies between different, but related context. A context dependency (X,c,Y) means that context Y only needs to be filtered if the condition c holds for context X. Optionally, a range d can also be specified for the values of this context parameter Y. For example, the domain-specific ontology used by the oNCS Application Component defines the dependency `(Person, hasDiagnosis some (hasAssociatedPathology some (hasSymptom galen:SensitivityToLight,)) LightIntensity)`. This dependency indicates that the oNCS Application Component is also interested in `LightIntensity` measurements when a patient is detected who is sensitive to light. Additionally, the domain-specific ontology associates each light sensitivity symptom with a threshold, e.g., 100 lm. Consequently, the range d of Y is defined as being greater than or equal to this threshold. This allows the oNCS Application Component to alert a caregiver when the light intensity is too high, i.e., greater or equal than 100, at the location of this patient.

Given the context dependencies, the filter rule generation algorithm works as follows. The `ContextToQuery` concept and the `hasContextDependency` relationship between `temporal:Entity` concepts are introduced in the *Upper Ontology*, as visualized in orange in Figure 3.2. To define a context dependency, a subclass of the `ContextToQuery` concept is defined with the necessary and sufficient condition that the ontological instance must have the relationship X `hasContextDependency` Y and $c \equiv true$. The `ContextToQueryLightSensitivePerson` concept at the bottom of Figure 3.2 illustrates this for the running light sensitivity example. Every time the context changes, the context dependencies are investigated by examining the membership of instances to the `ContextToQuery` concept through inferencing on the ontology. Once the membership is determined, the construction of the filter rules is straightforward. For example, consider that the oNCS Application Component is alerted that the patient in room 101 is diagnosed with a concussion, because his profile is updated in the database. The domain-specific ontology of the oNCS Application Component contains the knowledge that patients with a concussion have light sensitivity as a possible symptom. Consequently, the condition c of the context dependency holds and the reasoning inferences that this person is an instance of the `ContextToQueryLightSensitivePerson` concept. The filter rule is then constructed by defining an `Event`, which has as context the range Y of the `hasContextDependency`. To determine the device(s) from which this context Y should be filtered, the `ContextGroup` associated with the patient is analyzed. As previously explained, the `ContextGroup` represents a logical group of entities, e.g., a patients with his/her associated devices. For the running example, this results in the following filter rule:

Event **and** hasContext **some** (LightIntensity
and (hasValue **some** float[>=100])
and (isObservationOf **some**
(hasId **some** hasID 'L101')))

This rule filters light intensity observations from the room of the patient. For the published context to match with this filter rule, it must be of type `LightIntensity`. Moreover, its current value must be of type float and higher or equal than 100. Finally, the observation must be observed by a system with ID 'L101'. This is the ID of the light sensor in room 101. As previously indicated, this kind of static information is stored in databases, which can be queried by the application components. Note that the first example of the *Publishing context* Section matches with this filter rule. More information about this filter rule generation algorithm can be found in [58].

Distributed deployment: As the application components heavily depend on the SCB to receive relevant context information, the centralized design of the SCB forms a single point of failure and a performance bottleneck. However, the SCB can easily be distributed as it does not retain the published context data, i.e., only one event is processed at a time by the SCB. Different instances of the SCB, processing the various events in parallel, can thus easily be deployed without interfering with each other. An actual large-scale deployment of the SCB will thus most likely not correspond to a centralized physical process, but will be a virtual substrate distributed across the network. Two approaches can be used, which are compared in Figure 3.5.

On the one hand, the SCB can be replicated, meaning that each context dissemination instance will contain the same amount of filter rules and the published events are divided across the different SCBs and processed in parallel. This improves the scalability. Replication also removes the single point of failure as a

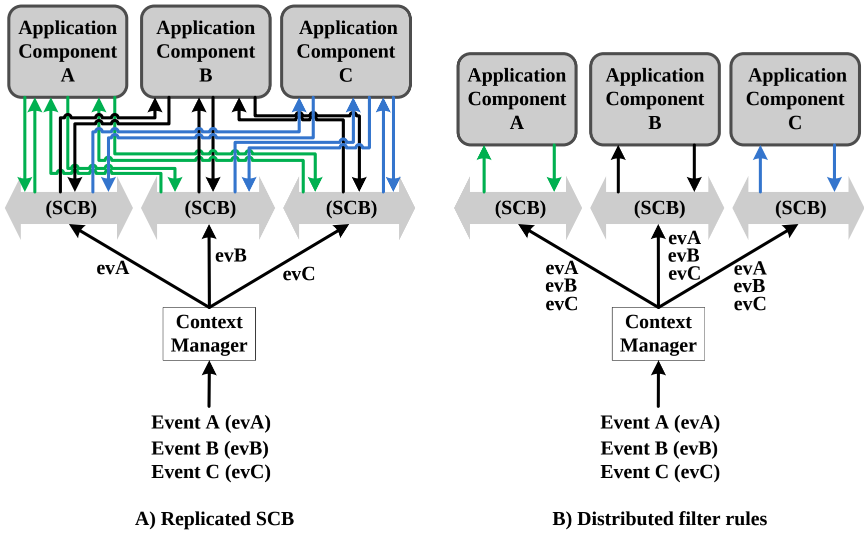


Figure 3.5: ***Comparison of the two approaches to distribute the SCB.*** *To prevent that the SCB becomes a single point of failure and to improve the performance, the SCB can be distributed. Approach A replicates the SCB, meaning that each context dissemination instance will contain the same amount of filter rules and the published events are divided across the different SCBs and processed in parallel. Approach B distributes the filter rules across different instances of the SCB, processing a published event in parallel.*

replicated instance of the SCB can easily replace a faulty one. This approach can be achieved by using the persistent team approach described in the Adaptive Agent Architecture [72].

On the other hand, the filter rules can be distributed across different instances of the SCB, processing a published event in parallel. In this case, the number of filter rules present in each context dissemination instance will thus be relatively low, which increases the throughput of each SCB significantly. Sophisticated algorithms have been proposed in literature to efficiently distribute the filter rules across the different peers [73]. This approach can be combined with the first to deal with faulty nodes.

3.3.4 Evaluation set-up

To evaluate the performance and benefits of the SCB, a prototype of the oNCS Application Component, extended with the Localization and Home Automation Application Components, was implemented and tested in the Patient Room of the Future (PRoF) [74]. PRoF is a high-fidelity mock-up of a near-future patient room integrating innovations from soft- and hardware developers as well as furniture.

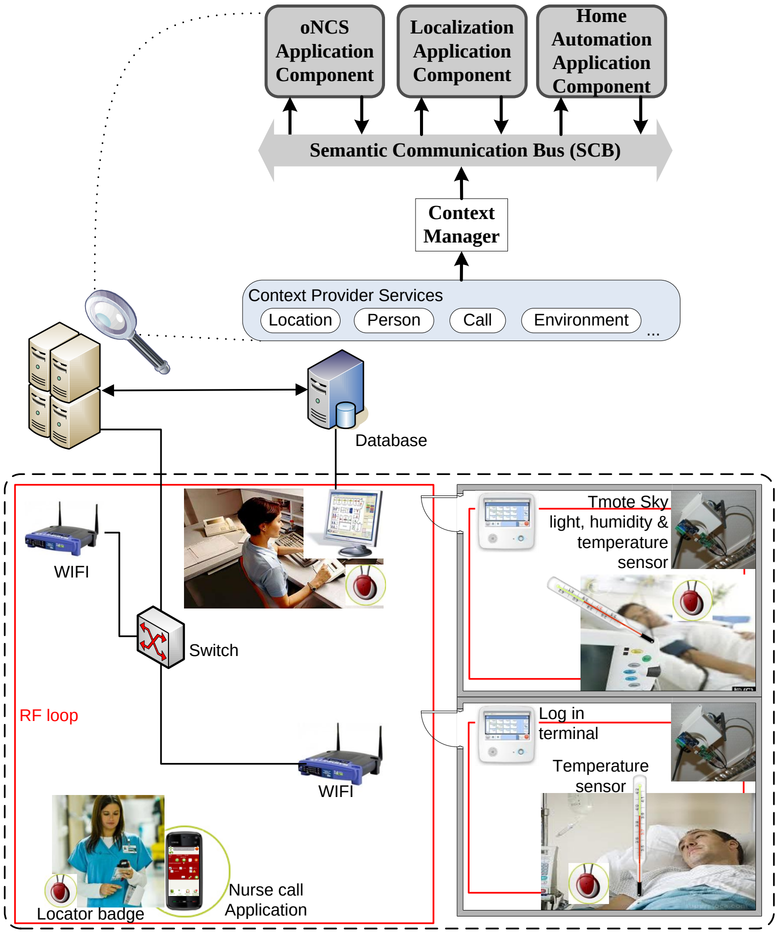


*Figure 3.6: **Deployment of the prototype in PRoF.** To evaluate the performance and benefits of the SCB, a prototype of the oNCS Application Component, extended with the Localization and Home Automation Application Component, was deployed in the Patient Room of the Future (PRoF). PRoF consists of a typical patient room and hallway found in a hospital setting, as well as a room mimicking a homecare setting. For the prototype, both rooms were equipped with a TMote Sky sensor board, which contains a light, temperature and humidity sensor. Also, each patient and staff member carried an RFID tag to track his/her location. Finally, each patient wore a bracelet that monitors the patient's body temperature. The developed prototype, consisting of the various context providers, the SCB, the oNCS, the localization and Home Automation Application Components, was installed in PRoF and integrated with the available nurse call and light control system, RF tags and sensors. Smartphones running the mobile nurse call application, which enable the nurses receive and answer calls, were also provided.*

The aim of PRoF is to make a patient feel more at home by exploring ways to put the patient and his needs first. PRoF consists of a typical patient room and hallway found in a hospital setting, as well as a room mimicking a homecare setting. For the prototype, both rooms were equipped with a TMote Sky [75] sensor board, which contains a light, temperature and humidity sensor. Also, each patient and staff member carried an RFID tag to track his/her location. Finally, each patient wore a bracelet that monitors the patient's body temperature. The developed prototype, consisting of the various context providers, the SCB, the oNCS, the localization and Home Automation Application Components, was installed in PRoF and integrated with the available nurse call and light control system, RF tags and sensors. Smartphones running the mobile nurse call application, which enable the nurses receive and answer calls, were also provided. Figure 3.6 visualizes the deployment of the prototype and accompanying sensors in PRoF. As PRoF contains only two rooms and the number of users was also limited, only one instance of the SCB was deployed.

This prototype allowed users to experience a fully immersed, contextual experience of the system in a lifelike context. As we wanted the participants to have a complete experience of the system, small groups were invited, i.e., two or three users per workshop, so that they would be occupied at all times and the researchers could follow them one-on-one. As such, seven workshops were organized for 15 participants. During a 2.5 hour role-playing workshop, the participants were asked to play out seven scenes. For each scene, a test user received a persona card and a context card, informing the test user of the role he or she would have to take up and what he or she would have to do. Afterwards, the functionalities of the system were elaborately discussed.

Sensor	Nr. of sensors	Nr. of observations per sensor	Total nr. of observations per hour
Light	1/room	1/sec	72,000
Temperature	1/room	1/sec	72,000
Humidity	1/room	1/sec	72,000
RFID tag	1/person	1/sec	144,000
Body temperature	1/patient	1/sec	108,000

Table 3.2: Sensor data generated in a department with 20 rooms, 30 patients and 10 staff members

However, as PRoF contains only two rooms, it was difficult to thoroughly evaluate the performance of the system based on these user tests. To mediate this, simulations were performed based on realistic data gathered from observations and interviews performed at Ghent University Hospital [76]. The simulated department contains 20 rooms, 30 patients and 10 staff members, who answer calls. It was again assumed that each room is equipped with a TMote Sky, the temperature of each patient is monitored with a bracelet and the location of all the staff members and patients is tracked. The simulation of the observations of the RF tags was based on realistic data gathered about the walking behavior of caregivers and patients in several departments of Ghent University Hospital. The simulation of the observations of the other sensors was based on stakeholder input. Table 3.2 gives an overview of the amount of data generated by each sensor for the simulations. As the goal of the simulations was to assess the performance and scalability of the SCB, only one instance was deployed.

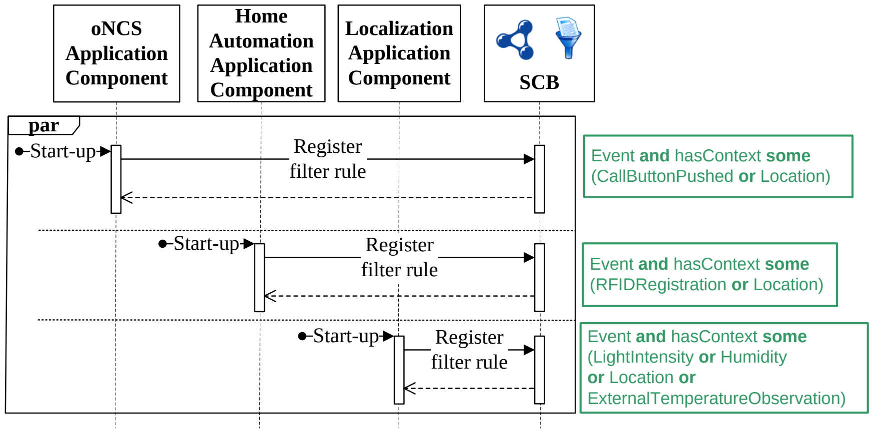


*Figure 3.7: **First step of the implemented scenario: Configuration.** This sequence diagram visualizes the different actions taken during the first step of the scenario realized by the implemented prototype. In this step, the different application components are configured by registering filter rules with the SCB such that they receive the context information that they are always interested in, independent of the current context. The filter rules are depicted in the squares on the right hand side of the figure. The "par" indicates that the different application components can perform their actions in parallel.*

The prototype is able to realize the example detailed in the *Introduction* Section. This scenario consists of the following steps. First, the oNCS, Localization Component and Home Automation Application Components register filter rules with the SCB, as visualized in Figure 3.7, to receive context information they are interested in, independent of the current context. Second, the application components work together to detect the presence of a nurse in a patient's room and turn on the light at the appropriate level. This is shown in Figure 3.8. Third, the application components receive information from the database indicating that a particular patient has a concussion. Consequently, the oNCS Application Component registers a filter rule to receive specific context information pertaining to the light intensity in this patient's room as illustrated in Figure 3.9. Next, a visitor enters the room, which causes the light to be automatically dimmed as this patient

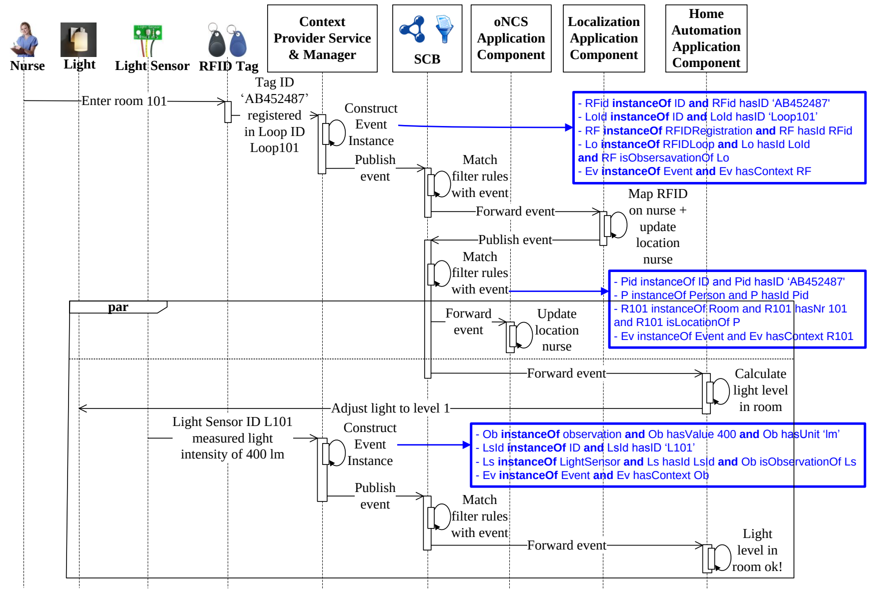


Figure 3.8: ***Second step of the implemented scenario: Turn on light when nurse enters the room.*** *This sequence diagram visualizes the different actions taken during the second step of the scenario realized by the implemented prototype. In this step, a nurse enters the room of the patient. This is detected by using RFID Tags, which send an Event to the SCB. The Localization Component is alerted of this Event as it matches with the filter rule this application component registered with the SCB. The location of the nurse is updated and this information is published on the SCB. This location information is forwarded to the oNCS and the Home Automation Application Components as it matches with their filter rules. The first just updates the location of the nurse in its local domain-specific ontology. The second uses this information to adjust the light level in the room of the patient. Finally, the light sensor in the room notices the change in light intensity and publishes this Event on the SCB. This Event matches with the filter rule of the Home Automation Component, which receives this Event and reasons that the light was adjusted to the appropriate level. The different Events are depicted in the squares. The "par" indicates that the different application components can perform these actions in parallel.*

has a concussion. The actions taken to realize this are similar to the ones visualized in Figure 3.8. However, as visualized in Figure 3.10, it remains possible for people to overrule the system and brighten the light in the room. An event similar to the last event indicated in Figure 3.8 is published on the SCB. However, because of the new filter rule, this event is not only forwarded to the Home Automation Component, but also to the oNCS Application Component. The Home Automation Component concludes that it cannot adjust the light level because it has been overruled. However, the oNCS Application Component also reasons on this event and alerts a nurse of the situation.

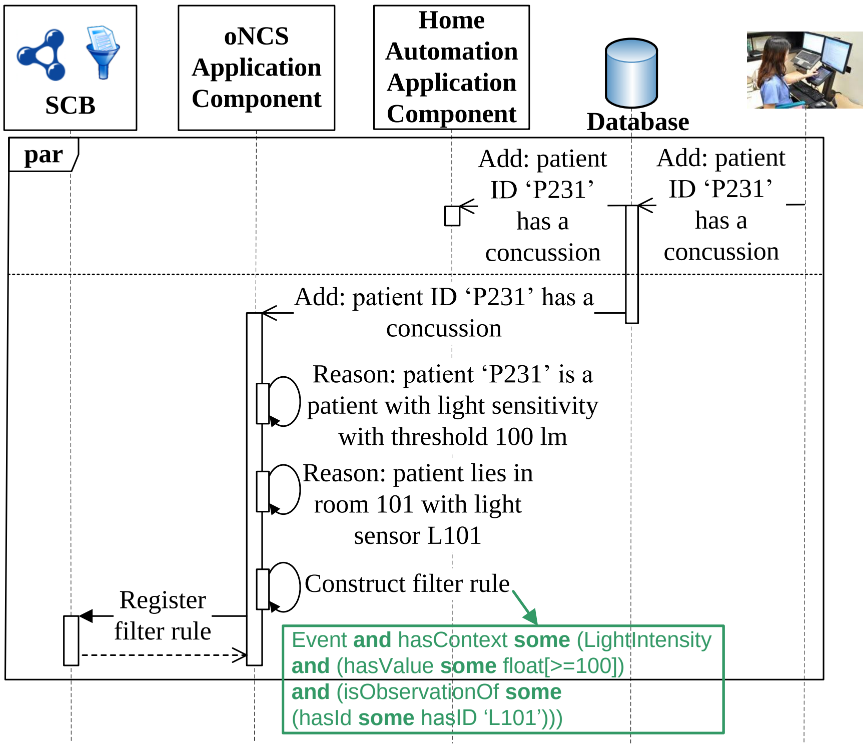


Figure 3.9: ***Third step of the implemented scenario: Registering a dynamic filter rule.*** *This sequence diagram visualizes the different actions taken during the third step of the scenario realized by the implemented prototype. In this step, the local domain-specific ontologies of the Home Automation and oNCS Application Components are updated to signal to these components that a particular patient has a concussion. The oNCS Application Component uses its local domain-specific knowledge to derive that this patient is sensitive to light. Consequently, the oNCS Application Component registers a filter rule to receive specific context information pertaining to the light intensity in this patient's room. The filter rules are depicted in the squares on the right hand side of the figure. The "par" indicates that the different application components can perform their actions in parallel.*

The evaluations were performed using the continuous care core ontologies needed to model the scenario as described in the *Continuous care ontology* Subsection. The core ontologies consist of 142 classes, 42 object properties, 21 data properties and 556 asserted axioms. The Protégé ontology editor [77, 78] was used to develop the ontologies in OWL-DL [20]. The context information published on the SCB and the filter rules registered by the application components are also expressed in OWL-DL. The prototype was built in Java, based on the Pellet OWL 2 reasoner [22] and the OWL Application Programming Interface (OWL-API) [79].

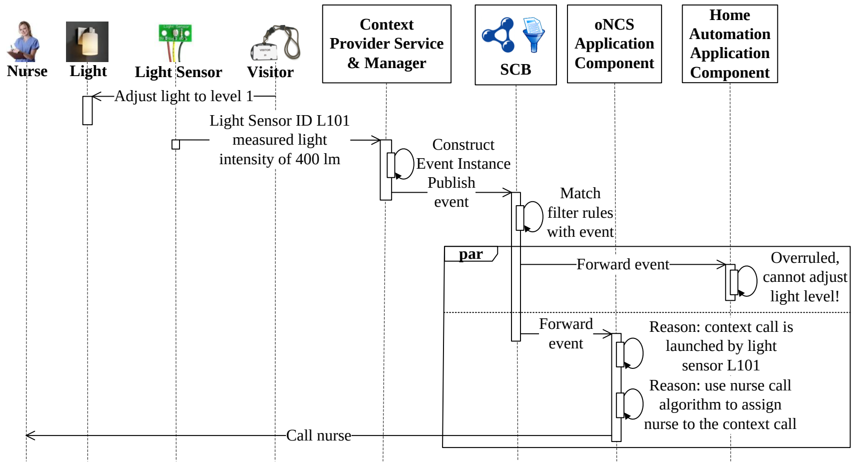


Figure 3.10: ***Fourth step of the implemented scenario: Visitor overrules the system and a nurse is alerted.*** *This sequence diagram visualizes the different actions taken during the fourth step of the scenario realized by the implemented prototype. In this step, a visitor adjusts the light level in the room and thus overrules the system. The light sensor in the room notices the change in light intensity and publishes this Event on the SCB. This Event matches with the filter rule of the Home Automation Component, which receives this Event and reasons that the light level cannot be adjusted to the needs of the patient as the component has been overruled. However, this Event also matches with the new filter rule that the oNCS Application Component registered in step 3 of this scenario. The oNCS Application Component derives that the light level is above the threshold for patients with a concussion and uses reasoning to find the most appropriate nurse to alert. The "par" indicates that the different application components can perform these actions in parallel.*

The cache was implemented using a combination of the Least-Frequently Used (LFU) and Least Recently Used (LRU) replacement algorithms, called the Least Recently/Frequently Used (LRFU) policy [80]. All the tests were performed under exactly the same conditions on the same isolated machine with following specifications: Advanced Micro Devices (AMD) Athlon 64 X 2 Dual Core Processor, 3000 megahertz (MHz) Central Processing Unit (CPU) and 2 Gigabyte (GB) of Random-Access Memory (RAM).

3.4 Results and discussion

Semantic reasoning and ontologies were adopted in this research to facilitate the exploitation and integration of heterogeneous context information delivered by the devices in an ambient-intelligent patient room. However, it is widely known that special attention needs to be paid to the number of instances in the ontology as this adversely affects the reasoning performance [81]. Consequently, if all the data from the ambient-intelligent environment would be delivered directly to the ontology-based application components, their performance would degrade drastically. By using the SCB, which only process one event at a time, filters can be defined to ensure that the application components only receive relevant context information. This way, the different infrastructure components are loosely coupled, effectively separating the devices delivering context information from the application components that process this information. Moreover, the SCB allows composing complex application components from a set of smaller application components, which perform specific reasoning tasks in parallel and forward their conclusions through the SCB.

Consequently, the effectiveness of the rules can be measured by the reduction in the amount of information that is processed by the application components as it is important that only the necessary context information is analyzed. However, the correctness of the rules is another important performance metric. The correctness is influenced by the amount of data that was wrongfully filtered. It is important not to filter too much information, as lack of information about the context might cause application components to take incorrect actions or no action at all. The goal is to increase the effectiveness, while maintaining the correctness. Consequently, the goal is to maximize the amount of data that is not forwarded, while ensuring that all the information that influences the actions of application components is correctly forwarded.

In the scenario detailed in the *Evaluation set-up* Subsection of the *Methods* Section, the SCB is capable of filtering a large amount of the simulated data from Table 3.2. As the Home Automation Component is only interested in sensor observations about the light intensity, the external temperature and the humidity, 54% of the generated sensor data is not forwarded to this component. Similarly, the Localization Component is only interested in RFID tag data, resulting in a 77% reduction of forwarded data. By taking advantage of the dynamic filter rule generation, none of the light intensity observations are forwarded to the oNCS Application Component if none of the patients in the department have light sensitivity symptoms. When the department does have such a patient, only 8.3% of all the light intensity observations are forwarded to the oNCS Application Component, assuming that it takes the nurse on average 5 minutes to respond to the context call, generated by the oNCS Application Component because the light intensity is too high in the room. This time interval was chosen based on observations and stakeholder expertise. This way, the filter rule generation process allows dynamically adapting the amount of data that is filtered based on the context. It can be noted that neither the oNCS nor the Home Automation Application Component ever receive RFID tag observations anymore. They depend on the Localization Application Component, which processes these raw RFID observations and publishes the resulting augmented location information. However, these location updates are far less frequent than the RFID tag observations, as only significantly changed locations are published, e.g., staff member is in another room.

The dissemination of new context consists of two steps. First, filter rules are created by the application components and registered with the Context Disseminator of the SCB. However, as filter rule registrations only occur occasionally, the introduced delay is negligible. Second, context is published to the SCB, matched with the filter rules and forwarded to the appropriate application component, if a match is found. The publication of context information happens frequently, as illustrated by Table 3.2. As such, it is important that events are matched with filter rules quickly and efficiently.

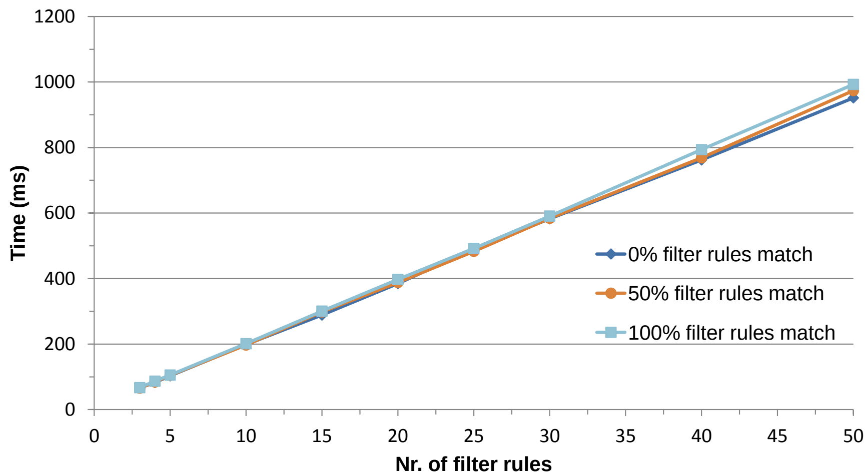


Figure 3.11: ***Average reasoning time as a function of the number of filter rules.*** *This graph visualizes the average reasoning time (y-axis in milliseconds (ms)) needed to publish, filter and forward one event as a function of the number of filter rules (x-axis) and the percentage of filter rules that match with this event, averaged over 30 iterations in case a cache miss occurs. The different lines indicate the amount of filter rules that match with the published event, namely 0%, 50% or 100%.*

When a cache miss occurs, the Context Disseminator has to reason on the information in the ontology to match the published event to the registered filter rules. The performance of filtering, matching and forwarding an event with the SCB as a function of the number of filter rules in case of a cache miss, is visualized in Figure 3.11. The lower and upper limits of the standard deviation are [2.19, 10.28], [2.58, 9.76] and [2.65, 10.07] when respectively 0%, 50% and 100% of the filter rules match with the published event. The graph shows that the processing time is linear in the amount of filter rules and that the influence of the percentage of filter rules that match the event is negligible. Note that for the described scenario, which contains at most 4 filter rules, an event is processed in on average 82.67

ms. However, the performance quickly decreases. For 50 filter rules, it takes on average 1 second to process an event if a cache miss occurs.

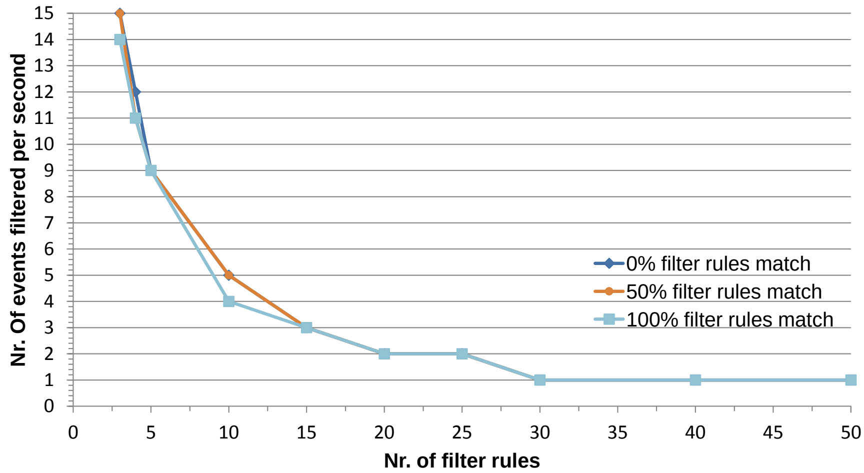


Figure 3.12: ***Number of events processed per second as a function of the number of filter rules.*** *This graph visualizes the number of events that can be filtered per second (y-axis) by one instance of the SCB as a function of the number of filter rules (x-axis) in case none of these events are present in the ache. This means that the Context Disseminator will have to reason on the ontology to filter, match and forward these events to the correct application components. An event is only counted when it has been completely processed and forwarded. The different lines indicate the amount of filter rules that match with the published event, namely 0%, 50% or 100%.*

It can be derived from Table 3.2, that every second all the sensors will output a new observation. In order to keep up with the publishing rate of the sensors (throughput), it is thus important that the SCB can process one event of each sensor in less than a second. Figure 3.12 visualizes the number of events that can be filtered per second by one instance of the SCB as a function of the number of filter rules in case none of these events are present in the cache. This means that the Context Disseminator will have to reason on the ontology to filter, match and forward these events to the correct application components. Note that the number of events on the y-axis is limited to those events which have been completely processed within the time limit, i.e., 1 second. Partially processed events are not considered. The graph shows that for the described scenario, which contains at most 4 filter rules, at most 15 events can be processed. If this result is mapped on the sensors in Table 3.2, this means that the SCB can process the events of 11 sensors of the simulated department, namely the events of 1 staff member (carrying an RFID tag) and 2 rooms (each fitted with 1 light, 1 temperature and 1 humidity sensor) with 2 patients in each room (each carrying an RFID tag and a body temperature sensor).

To enhance these performance results, three measures can be taken. First, the cache can be used. If the published event is present in the cache, a cache hit occurs and no reasoning needs to be performed to process the event and forward it to the appropriate application components. As the cache is implemented using LRFU, looking up whether an event is present in the cache and, in case of a cache hit, getting the filter rules and thus the application components to which the event should be forwarded can be performed with a worst-case performance of O(log n), where n is the number of entries in the cache [80]. Moreover, replacing an entry in the cache with a new entry in case of a cache miss also has a worst-case performance of O(log n). Consequently, looking up an entry in the cache performs much better than the reasoning that needs to be performed by the SCB when a cache miss occurs. It needs to be noted that the entries in the cache will become invalid when a new filter rule is added to the SCB as it is possible that the events already present in the cache now also match on this new filter rule. To resolve this issue, all the entries in the cache are flagged. When a cache hit occurs for a flagged entry, the Context Disseminator will perform the reasoning step, update the value in the cache and remove the flag. This way the whole cache will eventually be updated, either by performing reasoning or because flagged entries are replaced by new entries. To further speed up the runtime performance of the SCB, the cache could be filled with representative events during start-up. For example, the cache could be filled with room temperature events between 18 and 22° Celsius, as these are the most common room temperature values. However, this will significantly increase the start-up time of the platform and might only be a good choice when the filter rules do not change often.

As a second measure, the framework can be distributed. As explained in the *Flexible and semantic publish/subscribe mechanism* Section, two approaches can be used, namely replicating the SCB or distributing the filter rules across multiple instances of the SCB. A combination of both can also be used to ensure scalability as well as tolerance against faulty SCB instances. The choice of the distribution approach depends on the specific use case scenario. For use cases where the number of published events is high compared to the number of filter rules, the first approach is more suitable. As illustrated by Table 3.2, the use case presented in this paper falls into this category. A lot of events are published on the SCB, but only 4 filter rules are registered. The first approach is thus a good choice. As mentioned previously, this approach can be achieved by using the persistent team approach. The only parameter of this approach, is the minimum number of times R that the SCB should be replicated. To achieve a scalable deployment for the use case under scrutiny, R is set equal to the number of rooms, i.e., 20. As an SCB instance with 4 filter rules is able to process 15 events per second, the availability of at least 20 SCB instances guarantees that the 3 environment observations measured by the TMote Sky and the 4 patient observations, namely the location and temperature of the two patients, in each room per second can be efficiently processed. It also leaves enough processing power to process the location observations of the staff members and to allow the registration of additional context-dependent filter rules. In contrast, the second approach or the combination of both approaches is a better choice for use case scenarios with a lot of filter rules compared to the number of published events. Next to the parameter R, specifying the number of SCB instances that should be created, this approach also requires a specification of how the filter rules should be distributed across these instances. A simple algorithm consists of distributing the filter rules so that each instance of the SCB contains approximately the same amount of filter rules. This ensures that each instance of the SCB has a comparable performance.

Furthermore, it has been shown by the authors [58] that the performance of the SCB depends significantly on the complexity of the used core ontologies. It is therefore important to find a good balance between the desired throughput and the intelligence of the SCB, i.e., the expressiveness of the core ontologies and accompanying filter rules.

By balancing the expressivity of the core ontologies and the desired throughput, employing caches and distributing the SCB according to the specifics of the use case, the developed platform allows the realization of a plethora of healthcare scenarios in an efficient and scalable way.

To ensure the correctness of the filter rules it is important to continuously evaluate the platform with the domain experts, both during the development process and after the system has been deployed. Domain experts were constantly involved during the design and development of the continuous care ontologies, O'Care Platform and used application components to make sure the system correctly reflects the continuous work processes of the caregivers. Observations were performed to investigate which information is taken into account to perform a certain task or make a decision. A participatory methodology was used to develop the ontology and the accompanying algorithms. Moreover, the developed system has been thoroughly evaluated with the various stakeholders by allowing them to play scenarios in PRoF. If the system gets deployed, techniques can also be used to continuously monitor and improve the correctness. For example, situations can be examined in which the decision of the platform was overruled by the users and intermediate feedback can be gathered.

The user tests resulted in a considerable amount of feedback. However, the feedback mostly pertained to the user interface of the mobile nurse call application and the employed nurse call algorithm. No comments were made about missed or unnecessary calls, wrong light levels or the performance of the system.

3.5 Conclusions

In this article, a context-aware and pervasive framework was presented, capable of disseminating and filtering important care-related data of the different technologies available in an ambient-aware patient room towards a multitude of care applications, based on their information requirements. To realize this goal, the framework employs continuous care ontologies, which capture the information and knowledge being exchanged and utilized within healthcare settings. The applications can register, adapt and remove semantic filter rules on the fly to receive context information that is important to them at that moment. This way, the amount of data which needs to be processed by the applications is significantly reduced, which improves their performance and decreases overhead while maintaining an individualized approach. The strength of the platform is dependent on the correctness and completeness of the used ontologies. Therefore, a participatory ontology engineering methodology was designed which promotes user participation, while not overloading any of the involved stakeholders.

It was shown that the delay introduced by the context dissemination and filtering component is linear in the amount of filter rules and is negligible when 10 or less filter rules are registered. The performance of the platform can be significantly increased by employing a cache and by distributing the reasoning on the filter rules. The latter is achieved by replicating the context dissemination and filtering component or by distributing the filter rules across different instances of this component. A combination of both approaches can also be used. Moreover, the platform supports the composition of complex applications from a set of smaller applications in a loosely coupled manner. The simple applications perform specific reasoning tasks in parallel and notify their conclusions to other applications, which have expressed an interest in this kind of information.

3.6 Addendum

Figure 3.13 shows an example of how the Semantic Communication Bus (SCB) and the accompanying application components could ideally be deployed in a large hospital with different wards containing a large number of rooms. Some example applications are also indicated. The data generated by the equipment in each room can be managed by a separate SCB, as illustrated by SCB1 and SCB2 in Figure 3.13. If a room contains more than one patient, an SCB could even be used per patient. This reduces the amount of data that is processed per SCB. The applications that only process data pertaining to one patient only have to register their filter rules with the SCB that receives the data of this patient. This is illustrated by App A, B and C in Figure 3.13. As not all parameters are measured for each patient, this reduces the amount of filter rules per SCB. Moreover, the complexity

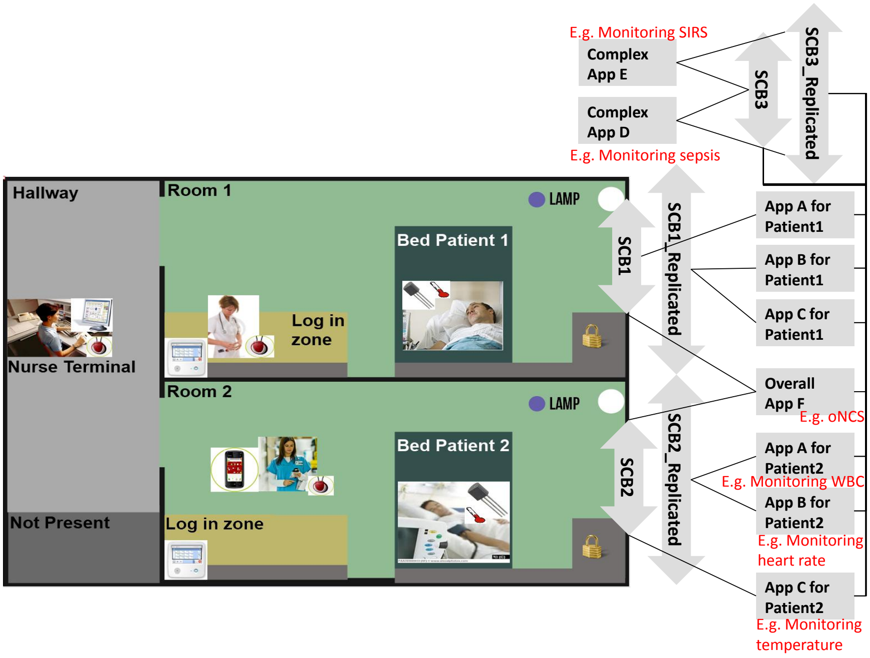


Figure 3.13: ***Example deployment of the O'Care Platform.***

of the filter rules can be reduced as the SCB only processes data pertaining to one room or patient. As shown by App F, the applications that need to process the data of different patients simultaneously to make an accurate conclusion can register their filter rules with the different room- or patient-based SCBs. The patient- or room-based applications can then forward their conclusions to yet another SCB, e.g., SCB3 in Figure 3.13. More complex applications that combine the conclusions of these different patient- or room-based applications can then register their filter rules with this SCB as visualized by App D en E. By replicating the SCBs, the filter rules can be further distributed amongst different SCBs. This further reduces the amount of filter rules per SCB.

Acknowledgment

We would like to acknowledge that part of this research was supported by the iMinds Project ACCIO co-funded by the IWT, iMinds and the following partners: Televic NV, Boone NV, Dominiek Savio Instituut and In-Ham. Part of this research was performed in the PRoF1.0 demo room, which was realized in Poperinge, Belgium by the PRoF consortium in 2010. F. Ongenae and J. Famaey would like to thank the IWT for financial support through their Ph.D. grant. S. Latré is funded by the Fund for Scientific Research Flanders (FWO-Vlaanderen). We thank all the participants in ACCIO for their valuable contribution to the project.

Competing interests

The authors declare that they have no competing interests

Author's contributions

FO carried out the study, developed the healthcare applications and ontologies described in this paper and drafted the manuscript. JF and SL participated in the development of the Semantic communication Bus and helped with the evaluation. SD, SV and AA participated in the case study and development of the ontologies. SV was also involved in the evaluation. PV and FDT supervised the study, participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

References

[1] C. Orwat, A. Graefe, and T. Faulwasser. *Towards pervasive computing in health care - A literature review*. BMC Medical Informatics and Decsion Making, 8(26):18, 2008.

[2] M. Tentori, D. Segura, and J. Favela. *Chapter VIII: Monitoring hospital patients using ambient displays*. In P. Olla and J. Tan, editors, Mobile Health Solutions for Biomedical Applications, volume 1, pages 143–158. Medical Information Science Reference;, Hershey, New York, USA, 1st edition, 2009.

[3] J.-C. Burgelman and Y. Punie. *Chapter II: Information, Society and Technology*. In E. H. L. Aarts and J. L. Encarnacao, editors, True Visions: The Emergence of Ambient Intelligence, volume 1, pages 17–35. Springer;, Berlin/Heidelberg, Germany, 1st edition, 2006.

[4] *The ACCIO Project* [online]. 2012. http://www.ibbt.be/en/projects/overview-projects/p/detail/accio-2.

[5] Y. Punie. *The future of ambient intelligence in Europe: the need for more everyday life*. Communication and Stratégies, 57:141–165, 2005.

[6] T. R. Gruber. *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2):199–220, 1993.

[7] A. K. Dey and G. D. Abowd. *Towards a better understanding of context and context-awareness*. In D. R. Morse and A. K. Dey, editors, Proc. of the CHI Workskhop on the What, Who, Where, When and How of Context-Awareness, The Hague, The Netherlands, 1–6 April 2000. New York, NY, USA: ACM Press.

[8] H. E. Byun and K. Cheverst. *Utilizing context history to provide dynamic adaptations*. Applied Artificial Intelligence, 18(6):533–548, 2004.

[9] J. Hong, E. Suh, and S. Kim. *Context-aware systems: A literature review and classification*. Expert Systems with Applications, 36(4):8509–8522, 2009.

[10] M. Baldauf, S. Dustdar, and F. Rosenberg. *A survey on context-aware systems*. International Journal of Ad Hoc and Ubiquitous Computing, 2(4):263–277, 2007.

[11] W. Xue and H. K. Pung. *Chapter 8: Context-Aware Middleware for Supporting Mobile Applications and Services*. In A. Kamur and B. Xie, editors, Handbook of Mobile Systems Applications and Services, volume 1, pages 269–304. CRC Press, Florida, USA, 1st edition, 2012.

[12] A. K. Dey, D. Salber, and G. D. Abowd. *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*. Human-Computer Interaction Journal, Special Issue on Context-aware Computing, 16(2–4):97–166, 2001.

[13] T. Strang and C. Linnhoff-Popien. *A context modeling survey*. In J. Indulska and D. D. Roure, editors, Proc. of the 6th International Conference on Ubiquitous Computing (UbiComp), Workshop on Advanced Context Modelling, Reasoning and Management, pages 31–41, Nottingham, UK, 7 September 2004.

[14] T. Gu, H. K. Pung, and D. Q. Zhang. *A service-oriented middleware for building context-aware services*. Journal of Network and Computer Applications (JNCA), 28(1):1–18, 2005.

[15] H. Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, University of Maryland, Baltimore County, 2004.

[16] P. Korpipää, J. Mantyjarvi, J. Kela, H. Keranen, and E.-J. Malm. *Managing context information in mobile devices*. IEEE Pervasive Computing, 2(3):42–51, 2003.

[17] L. O. B. S. Santos, R. P. Wijnen, and P. Vink. *A service-oriented middleware for context-aware applications*. In Proc. of the 5th International Workshop on Middleware for Pervasive and Ad hoc Computing, pages 37–42, Newport Beach, Orange County, CA, USA, 26–30 November 2007. New York, NY, USA: ACM Press.

[18] E. Kim and J. Choi. *A context-awareness middleware based on service-oriented architecture*. In J. Indulska, J. Ma, L. T. Yang, T. Ungerer, and J. Cao, editors, Proc. of the 4th International Conference on Ubiquitous Intelligence and Computing, pages 953–962, Hong Kong, China, 11–13 July 2007. Berlin/Heidelberg: Springer.

[19] M. Román, C. Hess, R. Cerqueira, R. H. Campbell, and K. Nahrstedt. *Gaia: A Middleware Infrastructure to Enable Active Spaces*. IEEE Pervasive Computing, 1:74–83, 2002.

[20] D. L. McGuinnes and F. Van Harmelen. *OWL Web Ontology Language overview* [online]. 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210.

[21] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK, 2003.

[22] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A Practical OWL-DL Reasoner*. Journal of Web semantics: Science, Services and Agents on the World Wide Web, 5(2):51–53, 2007.

[23] B. Motik, R. Shearer, and I. Horrocks. *Hypertableau Reasoning for Description Logics*. Journal of Artificial Intelligence Research, 36:165–228, 2009.

[24] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML* [online]. 2004. http://www.w3.org/Submission/SWRL/.

[25] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. *Jena: implementing the semantic web recommendations*. In S. Feldman, M. Uretsky, M. Najork, and C. Wills, editors, Proc. of the 13th international Conference on World Wide Web (WWW) - Alternate track papers & posters, pages 74–83, New York, NY, USA, May 17–20 2004. New York, NY, USA: ACM.

[26] N. Bricon-Souf and C. R. Newman. *Context awareness in health care: A review*. International Journal of Medical Informatics, 76(1):2–12, 2007.

[27] U. Varshney. *Chapter 11: Context-awareness in Healthcare*. In Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring, pages 231–257. Springer Science & Business Media, LLC, New York, NY, USA, 1st edition, 2009.

[28] J. Bardram. *Applications of context-aware computing in hospital work - Examples and design principles*. In Proc. of the Annual ACM Symposium on Applied Computing, pages 1574–1579, Nicosia, Cyprus, 14–17 March 2004. New York, NY, USA: ACM Press.

[29] M. B. Skov and R. Th. Hoegh. *Supporting information access in a hospital ward by a context-aware mobile electronic patient record*. Personal and Ubiquitous Computing, 10(4):205–214, 2006.

[30] S. Mitchell, M. D. Spiteri, J. Bates, and G. Coulouris. *Context-aware multimedia computing in the intelligent hospital*. In Proc. of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system, pages 13–18, Kolding, Denmark, 17–20 September 2000. New York, NY, USA: ACM.

[31] V. Stanford. *Beam me up, doctor McCoy*. IEEE Pervasive Computing, 2(3):13–18, 2003.

[32] M. A. M. noz, M. Rodríguez, J. Favela, A. I. Martinez-Garcia, and V. M. González. *Context-Aware Mobile Communication in Hospitals*. Computer, 36(9):38–46, 2003.

[33] K. Fishkin, M. Wang, K. P. Fishkin, and M. Wang. *A flexible, low-overhead ubiquitous system for medication monitoring*. Technical report, Intel Research Seattle, Technical Memo IRS-TR-03-011, 2003.

[34] C. Floerkemeier and F. Siegemund. *Improving the Effectiveness of Medical Treatment with Pervasive Computing Technologies*. In Proc. of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at International Conference on Ubiquitous Intelligence and Computing, Seattle, Washington, USA, 25 October 2003.

[35] I. Korhonen, P. Paavilainen, and A. Särelä. *Application of ubiquitous computing technologies for support of independent living of the elderly in real life settings*. In Proc. of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at International Conference on Ubiquitous Intelligence and Computing, Seattle, Washington, USA, 25 October 2003.

[36] A. Mihailidis, B. Carmichael, J. Boger, and G. Fernie. *An intelligent environment to support aging-in-place, safety, and independence of older adults with dementia*. In Proc. of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at International Conference on Ubiquitous Intelligence and Computing, Seattle, Washington, USA, 25 October 2003.

[37] P. de Toledo, S. Jimenez, F. del Pozo, J. Roca, A. Alonso, and C. Hernandez. *Telemedicine Experience for Chronic Care in COPD*. IEEE Transactions on Information Technology in Biomedicine, 10(3):567–573, 2006.

[38] B. Hu, B. Hu, J. Wan, M. Dennis, H.-H. Chen, L. Li, and Q. Zhou. *Ontology-based ubiquitous monitoring and treatment against depression*. Wireless Communications & Mobile Computing, 10(10):1303–1319, 2010.

[39] T. Suzuki and M. Doi. *LifeMinder: an evidence-based wearable healthcare assistant*. In M. Beaudouin-Lafon and R. J. K. Jacob, editors, Proc. of the ACM Conference on Human Factors in Computing Systems, pages 127–128, Seattle, Washington, USA, 31 March–5 April 2001. New York, NY, USA: ACM.

[40] B. Jansen and R. Deklerck. *Context aware inactivity recognition for visual fall detection*. In Proc. of the Pervasive Health Conference and Workshops, pages 1–4, Innsbruck, Austria, November 29 2006.

[41] N. Agoulmine, M. J. Deen, and M. M. L. Jeong-Soo. *U-health smart home: Innovative solutions for the management of the elderly and chronic diseases*. IEEE Nanotechnology Magazine, 5(3):6–11, 2011.

[42] V. F. S. Fook, S. C. Tay, M. Jayachandran, J. Biswas, and D. Zhang. *An ontology-based context model in monitoring and handling agitation behaviour for persons with dementia*. In Proc. of the 4th IEEE International Conference on Pervasive Computing and Commununications Workshops (PERCOMW), pages 560–564, Pisa, Italy, 13–17 March 2006. Washington, DC, USA: IEEE Computer Society.

[43] D. Zhang, Z. Yu, and C.-Y. Chin. *Context-aware infrastructure for personalized healthcare*. Studies in Health Technology and Informatics, 117:154–163, 2005.

[44] F. Paganelli and D. Giuli. *An ontology-based system for context-aware and configurable services to support home-based continuous care*. IEEE Transactions on Information Technology in Biomedicine, 15(2):324–333, 2011.

[45] T. Chin. *Technology Valued, but Implementing it into Practice is Slow* [online]. 2004. http://www.ama-assn.org/amednews/2004/01/19/bisb0119.htm.

[46] J. Anderston and C. Aydin. *Evaluating the Impact of Health Care Information Systems*. International Journal of Technology Assessment in Health Care, 13(2):380–393, 1997.

[47] J. H. Jahnke, Y. Bychkov, D. Dahlem, and L. Kawasme. *Context-aware information services for health care*. In T. Roth-Berghofer and S. Schulz, editors, Proc. of the 27th German Conference on Artificial Intelligence, Workshop on Modeling and Retrieval of Context (MRC), pages 73–84, Ulm, Germany, 20–21 September 2004. Aachen, Germany: CEUR.

[48] J. Ash, D. Sittig, K. Guappone, R. Dykstra, J. Richardson, A. Wright, J. Carpenter, C. McMullen, M. Shapiro, A. Bunce, and B. Middleton. *Recommended practices for computerized clinical decision support and knowledge management in community settings: a qualitative study*. BMC Medical Informatics and Decision Making, 12(1):6, 2012.

[49] I. Filali, F. Bongiovanni, F. Huet, and F. Baude. *A Survey of Structured P2P Systems for RDF Data Storage and Retrieval*. International Journal Transactions on Large-scale Data and Knowledge-Centered Systems III, Special Issue on Data and Knowledge Management in Grid and P2P Systems, 6790:20–55, 2011.

[50] F. Manola, E. Miller, and B. McBride. *RDF Primer* [online]. 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[51] A. Valls, K. Gibert, D. Sánchez, , and M. Bateta. *Using ontologies for structuring organizational knowledge in Home Care assistance*. Internation Journal of Medical Informatics, 79(5):370–387, 2010.

[52] F. Ongenae, F. De Backere, K. Steurbaut, K. Colpaert, W. Kerckhove, J. Decruyenaere, and F. De Turck. *Appendix B: overview of the existing medical and natural language ontologies which can be used to support the translation process*. BMC Medical Informatics and Decision Making, 10(3):4, 2011.

[53] A. L. Rector, J. E. Rogers, P. E. Zanstra, and E. van der Haring. *OpenGALEN: Open Source Medical Terminology and Tools*. In Proc. of the annual American Medical Informatics Association (AMIA) Symposium, page 982, Washington, DC, USA, 8–12 November 2003. American Medical Informatics Association (AMIA). http://www.opengalen.org/.

[54] J. A. Blake and M. A. Harris. *The Gene Ontology (GO) project: structured vocabularies for molecular biology and their application to genome and expression analysis*. Current Protocols in Bioinformatics, 23(7.2.1–7.2.9):1472–6947, 2008. http://www.geneontology.org/.

[55] M. D. Rodríguez, M. Tentori, J. Favela, D. Saldaña, and J.-P. García. *CARe: An Ontology for Representing Context of Activity-Aware Healthcare Environments*. In Proc. of the AAAI Workshop on Activity Context Representation: Techniques and Languages, San Francisco, CA, USA, 7–8 August 2011. Menlo Park, USA: AAAI Press.

[56] C. E. Kuziemsky and F. Lau. *A four stage approach for ontology-based health information system design*. Artificial Intelligence in Medicine, 50(3):133–148, 2010.

[57] J. Famaey, S. Latré, J. Strassner, and F. De Turck. *An Ontology-Driven Semantic Bus for Autonomic Communication Elements*. In R. Brennan, J. Fleck, and S. van der Meer, editors, Proc. of the 5th International Workshop on Modelling Autonomic Communication Environments (MACE), pages 37–50, Niagara Falls, Canada, 28 October 2010. Berlin/Heidelberg, Germany: Springer.

[58] J. Famaey, S. Latré, J. Strassner, and F. De Turck. *Semantic Context Dissemination and Service Matchmaking in Future Network Management*. International Journal of Network Management, 22(4):285–310, 2011.

[59] S. Verstichel, E. De Poorter, T. De Pauw, P. Becue, B. Volckaert, F. De Turck, I. Moerman, and P. Demeester. *Distributed ontology-based monitoring on the IBBT WiLab.t infrastructure*. In T. Magedanz, A. Gavras, N. H. Thanh, and

J. S. Chase, editors, Proc. of the 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), pages 509–525, Berlin, Germany, 18–20 May 2010. Berlin/Heidelberg, Germany: Springer.

[60] R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger. *Evaluation techniques for storage hierarchies*. IBM Systems Journal, 9(2):78–117, 1970.

[61] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, USA, 2009.

[62] K. Kotis and G. A. Vouros. *Human-centered ontology engineering: The HCOME methodology*. International Journal of Knowledge and Information Systems (KAIS), 10(1):109–131, 2006.

[63] E. Simperl, M. Mochol, and T. Bürger. *Achieving Maturity: the State of Practice in Ontology Engineering in 2009*. International Journal of Computer Science and Applications, 7(1):45–65, 2010.

[64] M. Gruninger and M. Fox. *Methodology for the design and evaluation of ontologies*. In Proc. of the International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 19–20 August 1995. San Mateo, USA: Morgan Kaufmann.

[65] M. Uschold and M. King. *Towards a methodology for building ontologies*. In Proc. of the International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 19–20 August 1995. San Mateo, USA: Morgan Kaufmann.

[66] M. Fernández, A. Gómez-Pérez, and N. Juristo. *Methontology: from ontological art towards ontological engineering*. In Proc. of the American Asociation for Artificial Intelligence (AAAI) Spring Symposium Series on Ontological Engineering, pages 33–40, Stanford, USA, 24–26 March 1997. Menlo Park, USA: AAAI Press.

[67] H. S. Pinto and J. P. Martins. *Ontologies: how can they be built?* Knowledge and Information Systems, 6(4):441–464, 2004.

[68] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, A. Jacobs, M. Van Gils, A. Ackaert, S. De Zutter, P. Verhoeve, and F. De Turck. *Participatory Design of a Continuous Care Ontology: Towards a User-Driven Ontology Engineering Methodology*. In J. Filipe and J. L. G. Dietz, editors, Proc. of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, 26–29 October 2011. ScitePress Digital Library.

[69] M. J. O'Connor and A. K. Das. *A lightweight model for representing and reasoning with temporal information in biomedical ontologies*. In A. L. N. Fred, J. Filipe, and H. Gamboa, editors, Proc. of the International Conference on Health Informatics (HEALTHINF), pages 90–97, Valencia, Spain, 20–23 January 2010. INSTICC Press.

[70] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. *OWL-S: Semantic Markup for Web Services* [online]. 2004. http://www.w3.org/Submission/OWL-S/.

[71] F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck. *An ontology-based nurse call management system (oNCS) with probabilistic priority assessment*. BMC Health Services Research, 11(26):28, 2011.

[72] S. Kumar, P. R. Cohen, and H. J. Levesque. *The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams*. In Proc. of the 4th International Conference on Multi-Agent Systems, pages 159–166, Boston, MA, USA, 10–12 July 2000. IEEE Computer Society.

[73] C. Chen, C.-L. Tsai, and S.-J. Horng. *Exploiting attribute popularity distribution skew to enhance the performance of peer to peer publish/subscribe systems*. International Journal of Innovative Computing, Information and Control, 7(7):4047–4066, 2011.

[74] T. P. Consortium. *PRoF: Patient Room of the Future* [online]. http://www.prof-projects.com/.

[75] Moteiv Corporation. *TMote Sky datasheet* [online]. 2006. http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.

[76] *Ghent University hospital* [online]. http://www.healthcarebelgium.com/index.php?id=uzgent.

[77] T. H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, Proc. of the 3rd International Semantic Web Conference, pages 229–243, Hiroshima, Japan, 7–11 November 2004. Berlin/Heidelberg, Germany: Springer.

[78] Stanford Center for Biomedical Informatics Research. *The Protégé ontology editor* [online]. http://protege.stanford.edu/.

[79] M. Horridge and S. Bechhofer. *The OWL API: A Java API for OWL Ontologies*. Semantic Web Journal, Special Issue on Semantic Web Tools and Systems, 2(1):11–21, 2011.

[80] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim. *LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies*. IEEE Transactions on Computers, 50(12):1352–1361, 2001.

[81] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan, and S. Liu. *Towards a Complete OWL Ontology Benchmark*. In Y. Sure and J. Domingue, editors, Proc. of the 3rd European Semantic Web Conference, pages 125–139, Budva, Montenegro, 11–14 June 2006. Berlin/Heidelberg, Germany: Springer.

4

An Ontology-Based Nurse Call Management System (oNCS) with Probabilistic Priority Assessment

F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck.

Published in BMC Health Services Research, Volume 11, Issue 1, Pages 26, February 2011.

"Healthcare is treating us as averages, not unique individuals, [but] at the end of the day, the patient is not the same thing as the population."

– Eric Dishman (2009)

To thoroughly demonstrate how intelligent and performant healthcare services can be implemented to optimize continuous care using the developed continuous care ontology and O'Care Platform, a prototype of an ontology-based Nurse Call System (oNCS) was developed. This research thus corresponds to Research Contribution 5 discussed in Section 1.3 of Chapter 1. This chapter focusses on the application as a whole by presenting a) its general architecture, b) how the profiles of the staff members and patients are managed using the continuous care ontology, c) the novel nurse algorithm that takes the provided context information *into account to assign the most suitable caregiver to a call, and d) the implementation details. Moreover, simulation results are presented based on data gathered about one department at Ghent University Hospital. Finally, the performance of the novel nurse call algorithm is evaluated. More details about the probabilistic reasoning and threshold algorithm used to determine the priority of a call can be found in Appendix B. This appendix also contains the simulation results of two additional deparments at Ghent University Hospital. As a result of the feedback obtained during the evaluation of the prototype with the users as part of the participatory ontology engineering methodology discussed in Chapter 2, the nurse call algorithm was updated. The new version of the nurse call algorithm is detailed in Appendix A. This appendix also contains a description of the mobile nurse call application and summarizes the ten most important lessons learned pertaining to the development of a dynamic nurse call system.*

⋆ ⋆ ⋆

Abstract

Background: The current, place-oriented nurse call systems are very static. A patient can only make calls with a button which is fixed to a wall of a room. Moreover, the system does not take into account various factors specific to a situation. In the future, there will be an evolution to a mobile button for each patient so that they can walk around freely and still make calls. The system would become person-oriented and the available context information should be taken into account to assign the correct nurse to a call.

The aim of this research is (1) the design of a software platform that supports the transition to mobile and wireless nurse call buttons in hospitals and residential care and (2) the design of a sophisticated nurse call algorithm. This algorithm dynamically adapts to the situation at hand by taking the profile information of staff members and patients into account. Additionally, the priority of a call probabilistically depends on the risk factors, assigned to a patient.

Methods: The ontology-based Nurse Call System (oNCS) was developed as an extension of a Context-Aware Service Platform. An ontology is used to manage the profile information. Rules implement the novel nurse call algorithm that takes all this information into account. Probabilistic reasoning algorithms are designed to determine the priority of a call based on the risk factors of the patient.

Results: The oNCS system is evaluated through a prototype implementation and simulations, based on a detailed dataset obtained from Ghent University Hospital. The arrival times of nurses at the location of a call, the workload distribution of calls amongst nurses and the assignment of priorities to calls are compared for the oNCS system and the current, place-oriented nurse call system. Additionally, the performance of the system is discussed.

Conclusions: The execution time of the nurse call algorithm is on average 50.333 ms. Moreover, the oNCS system significantly improves the assignment of nurses to calls. Calls generally have a nurse present faster and the workload-distribution amongst the nurses improves.

4.1 Background

4.1.1 Introduction

Information technology is widely adopted in modern medical practice, especially to support administrative tasks, electronic patient records (EPRs) and data management [1, 2]. The challenge today is that several data sources and devices have to be manually combined and consulted by the staff members to take advantage of this information, even when carrying out one single task. This is a time consuming job [3]. An underdeveloped area of solution for this problem is the use of context-aware techniques to automatically exploit the medical information available to improve continuous care and personalize healthcare. This implies an emerging demand for the integration and exploitation of the heterogeneous information available from all the wireless devices, patient records and medical data. Building context-aware applications on top of an ontology can ideally do this. An important way to coordinate work, communicate and provide continuous care is by making use of a nurse call system.

The architecture of traditional place-oriented nurse call systems can be viewed in the left part of Figure 4.1. Each room has at least one button which can be used by the patient to call a nurse. All the buttons in a room are connected to a *Node*. All the *Nodes* of a department are connected with each other and a *Controller*. The *Controllers* are the heart of the system. They contain the intelligence to know what must happen when a call is made, for example which nurses must be called.

The *Nodes* can be divided into different departments which each have their own specific settings. Within a department, the *Nodes* can be further divided into different, possibly overlapping, nursing groups. Each group can have his own configuration settings concerning for example the priorities of the different kinds of calls. Each nurse, who is identified by his or her beeper or portable phone number inside the system, is assigned to at least one nursing group. A nurse will only receive calls of the nursing groups that this nurse is assigned to. The advantage of using nursing groups is that patients who need more attention can be equally divided amongst the groups or be put in a separate group to distribute the workload better amongst the nurses.

The traditional nurse call algorithm consists of predefined links of beeper or portable phone numbers to rooms. To make a call the patient pushes one of the fixed buttons in his room. All the beepers and portable phones of the nurses, who

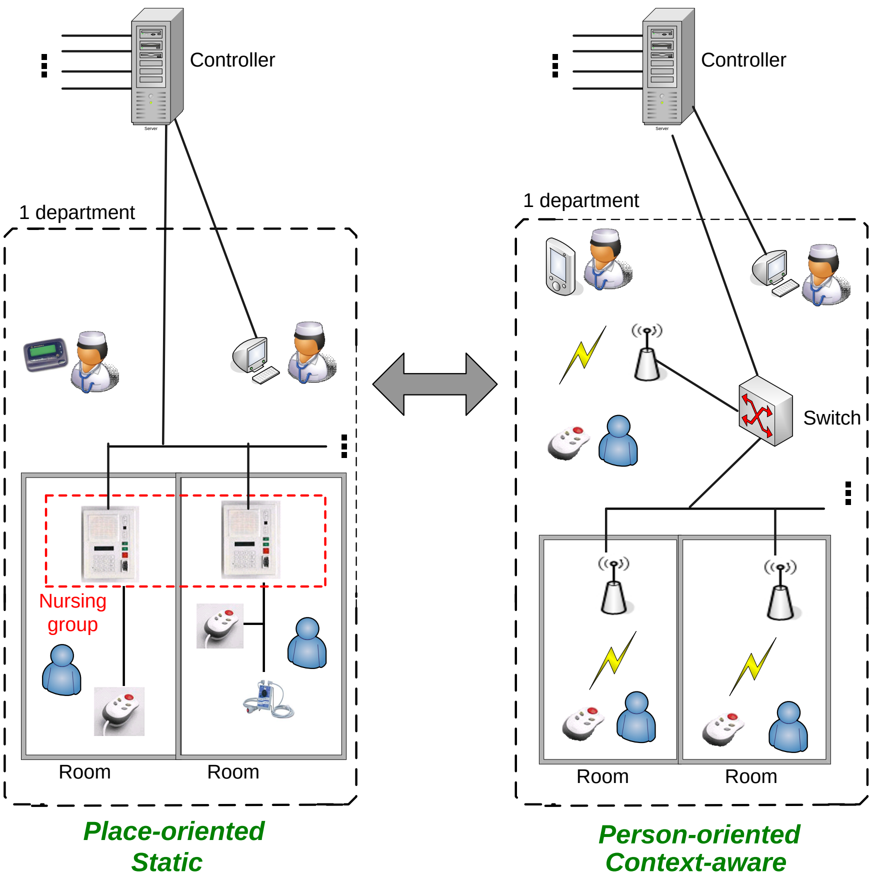


Figure 4.1: ***The traditional place-oriented and static nurse call system vs. the future person-oriented and context-aware approach.*** *The architecture of traditional nurse call systems can be viewed in the left part of the figure. Each room has at least one button which can be used by the patient to call a nurse. All the buttons in a room are connected to a Node. All the Nodes of a department are connected with each other and a Controller. The Controller has the intelligence to know what must happen when a call is made, for example which nurses must be called. A PC can be used to configure the controller. The nurses possess beepers or portable phone on which they can receive calls. Within a department, the Nodes can be further divided into different, possibly overlapping, nursing groups. A nurse will only receive calls of the nursing groups that this nurse is assigned to. The proposed architecture of the person-oriented and context-aware nurse call system can be viewed in the right part of the figure. Each patient has a mobile button so that they can walk around freely and still make calls. These calls are picked up by the sensor network and processed by the Controller. The Controller calls a nurse to handle the call. The nurse receives the call on his or her PDA.*

are in the nursing group that this room belongs to, are activated. The nurses decide on their own if they are going to interrupt their current task to answer the call or not. The nurse who reaches the room first will handle the call.

On one hand, the current nurse call systems are place-oriented. When a patient makes a call with a button that is fixed to a wall of a room, the called nurse simply goes to the room where the call came from. Herewith two important assumptions are made: the patient must still be in the room and it must be the patient who lies in the room that made the call. A patient can also only make calls inside his room. It is dangerous to become unwell, e.g. heavy respiratory or heart problems, inside a hallway, staircase or outside. This leads to patients being confined to their room to ensure their safety.

On the other hand, the system does not take into account various factors specific to a situation, such as the risk factors of a patient or the characteristics of the staff. Multiple nurses, namely all the nurses inside the nursing group that this room belongs to, are called. They have to decide for themselves if they are going to interrupt their current task to answer the call. They have no information about the priority and the kind of call or about the patient to guide them in this decision. If they interrupt their current work, which can also be a call, they have to remember themselves that they have to return to it. It is possible that more than one nurse goes to answer the call. This makes the whole system somewhat unreliable and inefficient.

Overall there is a transition to a world with more mobile and wireless devices [4]. In a study of Miller [5] the user friendliness and influence on nursing time is compared of two nurse call systems. The first system is comparable to the nurse call system detailed above. In the second system the staff members additionally were given locator badges through which they could be constantly tracked. 80% of the participants in the study preferred the second system to the first one. This is because a lot of time in a hospital is spent on *trying to find someone*. This claim is supported by a study of Linden [6] which found that almost 10% of nursing time is spent *looking for someone*. By using the locator badges this became an easier and less time-consuming task. Thus in the future, there will be an evolution to a mobile button for each patient so that they can walk around freely and still make calls, as can be seen in the right part of Figure 4.1.

This evolution implies a lot of changes, for example the nurse has to go to the exact location of the patient and the patients can make calls from anywhere in or outside the hospital. This huge impact is comparable to the introduction of the mobile phone. In the past we used to call to a telephone (a place) and ask for the correct person. Now we call a mobile phone en we immediately expect to have the right person on the line. Context information becomes increasingly important in a world with more and more wireless devices that have to be in touch with the environment around them. Lots of problems in current nurse call system are caused by the fact that they do not take the context information into account. The study of Linden also found that nurses are often called for tasks that could also be done by a less qualified staff member. Another study by Miller [7] supports this claim by concluding that on average 51% of the time registered nurses perform activities outside their role definition and do not require their level of knowledge and ability. Rerouting these kinds of calls to other staff members might greatly improve response time and patients satisfaction. Studies of call light use have also found that a large amount of calls are accidental calls [8]. Finding a way to indicate these calls might greatly improve the work pressure put on nurses and caregivers. Some features of the nurse call system which were identified as favorable to the performance of the staff are: locating staff, direct room-to-room communication and identification of the importance of calls, e.g. accidental or not or specifying condition and history of the patient.

In this article a novel software platform, the *ontology-based Nurse Call System (oNCS)*, is proposed that supports the transition to mobile and wireless nurse call buttons. Additionally, this platform efficiently manages the profiles of the staff members and the patients by encoding this context information into an ontology [9]. A new nurse call algorithm was developed that dynamically adapts to the situation at hand by taking the profile information into account such as the location and the characteristics of the staff and the patients, the current tasks of the staff members and the priorities of the calls. All this information is used to find the best staff member to handle a specific call and thus eliminate the above mentioned problems currently present in nurse call systems.

To clearly illustrate the person-oriented nature of the platform, the context information about the risk factors of a patient is used to dynamically determine the priority of the call this patient is making. By using probabilistic reasoning algorithms, the probability that a specific call made by a specific patient has a certain priority can be determined. These probabilities are derived from the different risk factors this patients has because they will influence the probability that a patient makes urgent calls. All these probabilistic values are combined in an intelligent manner to determine the most suitable priority for this call.

4.1.2 Objectives

The aim of this research is the design of a software platform that enables the transition to mobile and wireless nurse call buttons in hospitals and nursing homes and employs an intelligent nurse call algorithm that takes the profiles of the staff members and patient into account. The platform should offer the advanced features listed below:

- *Profile management:* In order to achieve a nurse call algorithm that adapts to the situation at hand, context information about the profiles of patients and staff members should be managed efficiently.

- *Dynamic priority assessment:* Instead of statically defining the priority of a call in advance, it should depend on the profile of the patient and more specifically on his or her risk factors. As patients with a certain profile can still make calls of varying priority, this information should be modeled probabilistically. As it is difficult to accurately determine the exact probability with which a patient with a certain profile will make a call of a certain priority, the platform should be able to handle probabilistic intervals.

- *Mobile:* The platform should give the patients enough mobility. They should be able to wander around the whole hospital and a limited area outside of the hospital for example the smoking area and the parking lot. They should be able to make calls in all these areas without their call getting lost because of bad reception. The mobile buttons should also be easy to operate.

- *Location-Aware:* The platform should be able to detect the locations of patients and staff members in a sufficiently accurate way and take this information into account when finding a suitable staff member to handle a call. This data should be constantly monitored and transparently delivered to the system.

- *Efficient staff assignment:* The nurse call algorithm should ensure that an optimal matching is achieved between the profiles of the staff members and the profile of the patient, when finding a suitable staff member to handle a call. An efficient workload distribution should be achieved between all the staff members who can handle each type of calls. A good balance between safety and cost should be achieved. The quality of care may not be undermined.

- *Reliable:* Four kinds of faults can occur: the server can go down, a call is not delivered to the server, a call is not delivered to the PDA of the staff member or the location information cannot be received or is inaccurate. The platform has to be able to cope with each of these situations. Calls may never be lost and it should always be able to call at least one staff member. A good logging infrastructure is needed to ensure that it is always known which patients made calls, which staff members handled them and how long it took until a staff member was at the location.

- *Performance:* The performance of the platform and the algorithms should be such that general guidelines can be imposed, for example, the guideline that stipulates that at least one staff member should arrive at the location of the patient within 3 minutes when an urgency call was made and within 5 minutes for other calls. As these time constrictions include walking to the patient, the time needed by the algorithm to assign a suitable staff member to a call should be negligible.

- *Generic:* It should be possible to plug-in new components, independent of implementation languages, operating systems and hardware by providing generic interfaces. New applications to visualize and input information from and into the platform should be easy to develop and plugged into the system.

- *Scalability:* The platform should be able to handle to large amount of profile information that is available about all the staff members and patients currently in the hospital. It should also be able to handle the large amount of calls that can daily enter the system.

4.1.3 Related work

On one hand, general purpose frameworks and models have been proposed that capture general concepts about contexts in an ontology and provide reasoning on this contextual model. For example, In Preuveneers et al. [10] an adaptable and extensible ontology is proposed for creating context-aware computing infrastructures, ranging from small embedded devices to high-end service platforms. In Gu et al. [11] an OSGi-based infrastructure for context-aware applications is proposed and Chen et al. [12] defined a context ontology based on OWL to support ubiquitous agents. However, all these frameworks are not specific for the healthcare domain.

On the other hand, many ontologies have been developed for the healthcare domain to model context, mainly for medical decision making [13, 14]. However, some ontologies that address the continuous care context have also been developed. For example, the ontology *OntHos* [15] was developed to model hospital scenarios and to facilitate their interoperability and Kataria et al. [16] implemented an ontology for an intelligent hospital ward to address data sharing and semantic heterogeneity. However, these papers do not address the context-aware reasoning that should take place on top of the ontology.

Yao et al. [17] tried to fill the gap between general purpose context-aware frameworks and a healthcare domain specific ontology. They propose the *CIHO* model, an extensible hospital ontology to represent, manipulate and access hospital information in intelligent environments. Additionally, they present examples of ontology reasoning and rule-based reasoning to show how context-aware services can be built. However, no complete service was built and evaluated.

In this paper we build further on the work of Yao et al. to unite the research on ontologies for continuous care with the research on frameworks for context-aware applications. A general purpose context-aware framework, namely the *Context-Aware Service Platform (CASP)* [18], is extended with a continuous care ontology which models the profile information of staff members and patients and context information about tasks and nurse calls. The main contribution of our work is the incorporation of probabilistic information in the ontology and the development of sophisticated probabilistic reasoning algorithms to achieve a sophisticated context-aware application. Additionally, the novel nurse call system was thoroughly evaluated through simulations based on realistic data.

4.1.4 Paper organization

The remainder of this paper is organized as follows. The *Methods* Section starts with a general description of the platform. Secondly, it is detailed how the profiles of the staff members and patients are managed by employing an ontology. It is also explained how information about the priorities of calls can be modeled so that it depends probabilistically on the risk factors of patients. Thirdly, the developed algorithms are presented. It is detailed how the probabilistic information can be used to determine the priority of a call. An overview of the novel nurse call algorithm is given that takes all the profile information in the ontology into account to find the best staff member to handle a call. The fourth subsection describes the implementation details. To test and demonstrate the advantages and performance of the system, a simulation was set up with realistic data provided by Ghent University Hospital [19]. The set-up is detailed in the final subsection of the *Methods* Section, while the results are discussed in the *Results* Section. The *Discussion* Section presents a critical discussion of the platform and its benefits. Finally, the main conclusions are highlighted in the last Section.

4.2 Methods

4.2.1 General concept

The main functionality of the person-oriented nurse call system with probabilistic risk assessment is to provide efficient support for wireless nurse call buttons and to employ a more sophisticated nurse call algorithm that takes the profiles of the staff members and patients into account. The general concept of the platform is illustrated in Figure 4.2.

Patients can walk around freely in the hospital with their wireless nurse call buttons. These buttons periodically broadcast a message which is picked up by the nearby sensors. The large number of available sensors guarantees that another sensor can pick up the message in case the closest one is malfunctioning. This information then travels through the switch to the back-end server, as can be seen in the bottom part of Figure 4.2. Existing state-of-the-art algorithms [20, 21] can be used to detect the accurate location of the patient out of this information by taking, for example, the signal strength perceived by the various sensors into account.

When the location cannot be calculated or is inaccurate, the previous location information is used until the next broadcast is detected. When the patient makes a call, a call message is sent in a similar manner. In this case the server does not only update the location of the patient, but also initiates the algorithm to find the most appropriate staff member to handle the call. The location of the patient is updated and monitored until a staff member is at the scene to handle the call.

Each staff member has a PDA which provides a user-friendly *Graphical User Interface (GUI)*. Information about the patients such as their risk factors or location can be requested. The PDA also notifies the staff member of calls that this staff member has been assigned to. The staff member is able to request information about the call such as where it originated from and what the priority is. The staff member can also indicate if he/she is going to handle the call or not. The sensor network is used to automatically detect that the staff member is at the location of the patient and is thus handling the call.

A desktop is available in each department which provides the head nurse with a *GUI* to input and visualize information about the department. The head nurse can input information about the patients, such as their risk factors or which rooms they occupy, and about the staff members, such as their characteristics or the patients they are responsible for. Information about the department is displayed in an overview window which shows which nurse has been assigned to which patient and where all the staff members and patients currently are. By clicking on a staff member or patient, the head nurse can view additional information about this person.

The new *ontology-based Nurse Call System (oNCS) platform* handles all the communication to and from these devices. The platform contains an ontology which is used to model all the profile information about the patients and staff members. The platform offers a wide range of *Web Service* [22] methods to transparently gain access to this information. Transparent access means that applications or users, who want to input data into the *oNCS system* or extract data from it, do not have to be aware of the underlying structure of the data e.g. the ontology or database. The Web Service provides an interface to input or extract data from the system, while the translation to the correct ontology or database query is kept completely hidden. This *Web Service* can be called from anywhere in the network. The *Provider Services* transform the inputted information to data that can be inserted in the ontology. The *Query Services* transform the data from the ontology to information that can be processed by the applications on the PDAs or desktops. These generic *Web Services* make it easy to write and plug new applications into the platform. This is further detailed in *The oNCS platform* Subsection of the *Implementation details* Section.

The ontology contains all the necessary context information about the hospital such as information about the profiles of the staff members, the profiles of the

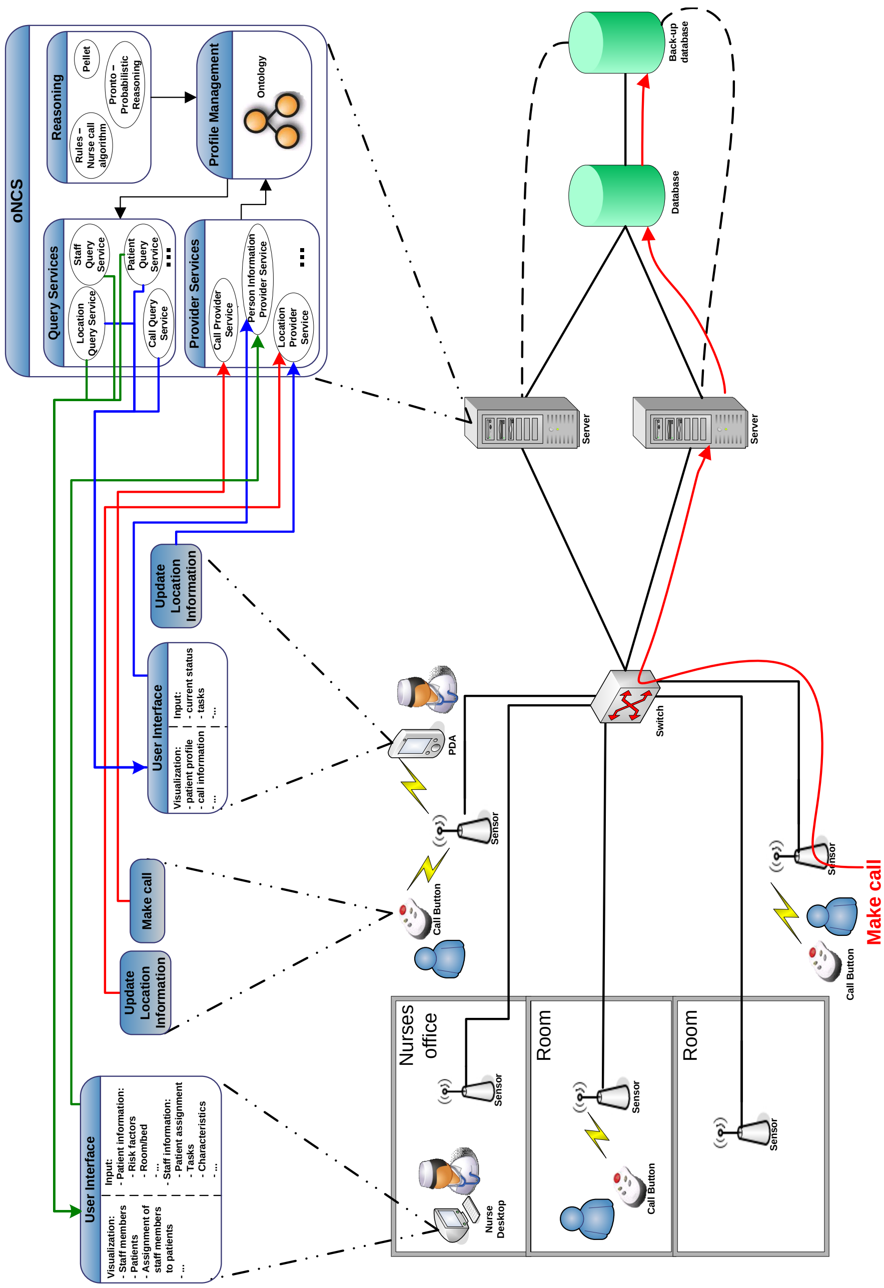


Figure 4.2: ***General concept of the oNCS platform with probabilistic risk assessment and profile management.***

patients and the calls. It also contains information about the risk factors of the patients. General information about the priorities of calls is modeled with probabilistic intervals in the ontology. These priorities thus depend probabilistically on the risk factors of the patients. The ontology is further detailed in the *Profile Management* Section.

Rules implement the novel nurse call algorithm that takes all the information in the ontology into account to find the best staff member to handle a call. The matching of a staff member to a call is not solely based on the fact that this staff member is responsible for the patient. Additional information such as the location of the staff members and the patient, the priority of the call, the characteristics of the staff member and the patients and the current task of the staff member are taken into account. The *Rules* are automatically triggered when a new call is inserted into the ontology. As a result the call is send to the PDA of the staff member who has been chosen to handle it. To ensure the reliability of the system, the algorithm also contains a time-out procedure. When a staff member has not indicated that he/she is going to handle the call within a certain amount of time, the call is launched again. The algorithm is further explained in *The nurse call algorithm* Subsection of the *Algorithms* Section.

The priority of a call is determined by reasoning algorithms that reason on the probabilistic information in the ontology about the risk factors of a patient. This priority can then be taken into account in the nurse call algorithm. The probabilistic reasoning algorithms are detailed in the *Priority Assessment of a call* Subsection of the *Algorithms* Section.

4.2.2 Profile management

In order to achieve a nurse call algorithm that adapts to the situation at hand, context information about the profiles of patients and staff members should be managed efficiently. Ontologies can be used to structure and represent knowledge about a domain in a formal way [9]. This knowledge can then easily be shared and reused. Because of the foundation of ontologies in *First-Order Logic (FOL)*, the models and description of the data in these models can be formally proofed. It can also be used to detect inconsistencies in the model as well as infer new information out of the correlation of this data. This proofing and classification process is referred to as reasoning.

To develop the *oNCS ontology*, a couple of concrete situations were studied in cooperation with the experts in the domain of nurse call systems at Televic NV [23]. For each situation the relevant context information was extracted and the ontology was augmented with it. It took several iterations and meetings with domain experts to get the desired ontology [24, 25]. The subsections below highlight the most important parts of the ontology.

4.2.2.1 Profile model of the staff members and patients

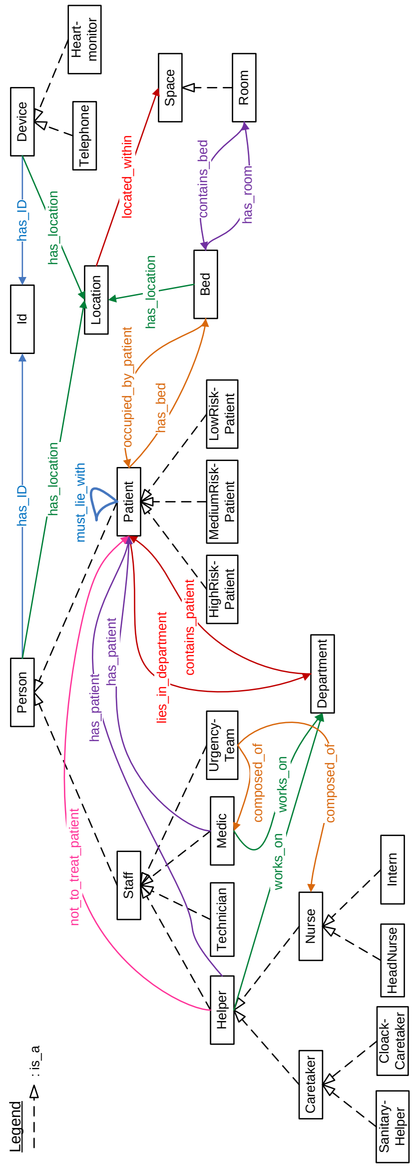


Figure 4.3: ***Fragment of the ontology that models the context information about the staff members and patients.*** *Fragment of the ontology which models the patients and staff members of the hospital who can answer calls. The squares represent the classes. The arrows with the striped lines indicate subclass relationships. The other arrows and lines indicate relations between classes (object properties).*

First, the patients and the staff members of the hospital who can answer calls were modeled, as can be seen in Figure 4.3. The current location is tracked for each staff member and patient. All staff members have associated beepers and/or portable phone numbers. It is also modeled on which departments a staff member works and on which department a patient lies. Some information is also maintained for administrative purposes such as names, IDs, beds and rooms.

Helpers can have different specializations. Two special types of nurses, namely head nurses and interns, and caretakers have been defined. Sanitary helpers are responsible for *caring* tasks such as cleaning a bed or fluffing a pillow. Family caregivers are volunteers.

In the place-oriented system, each helper was associated with a nursing group. However, in the person-oriented system it is more logical to associate each helper with a group of patients for whom this helper is responsible. This makes the system very flexible, as these groups can be dynamically adapted to equally divide the work load among the different helpers. Each medical staff member is also responsible for one or more patients.

Some characteristics about the helpers are modeled, which can be seen in Figure 4.4. For the current simulations, the following classes were used: which languages the helpers speak, their gender, their nationality and their religious beliefs. Helpers can indicate patients that they do not want to treat. Patients can then indicate which characteristics they would prefer to be present in the helper that treats them. So, patients cannot directly indicate that they do not want to be treated by a particular helper.

It can be indicated if a patient has one or more risk factors. A complete list of risk factors could be constructed based on a thorough study of the risk factors of patients and the reasons for the calls that they make. Unfortunately, such studies have not been conducted to the knowledge of the authors. To highlight the possibilities of the system, a (not exhaustive) list of risk factors was assembled by experts from both the medical and nurse call domain, as can be seen in Figure 4.4.

When a patient exhibits a risk factor, he is assigned a probability of belonging to a risk group namely `High`, `Medium` and `Low Risk Patients`. To give a preliminary idea of the benefits of this system, the probabilities were determined by domain experts. At it is difficult to determine exact probabilities for these cases, probabilistic intervals were employed. For example, a diabetic patient has at least 50% chance of being a high risk patient. This is encoded as the probabilistic interval [0.5,1] in the ontology.

Off course patients can have several risk factors, in this case the system will reason over the different probabilities to determine the general probability that a patient belongs to a risk group. This reasoning process is explained in more detail in the *Priority assessment of a call* Subsection of the *Algorithms* Section.

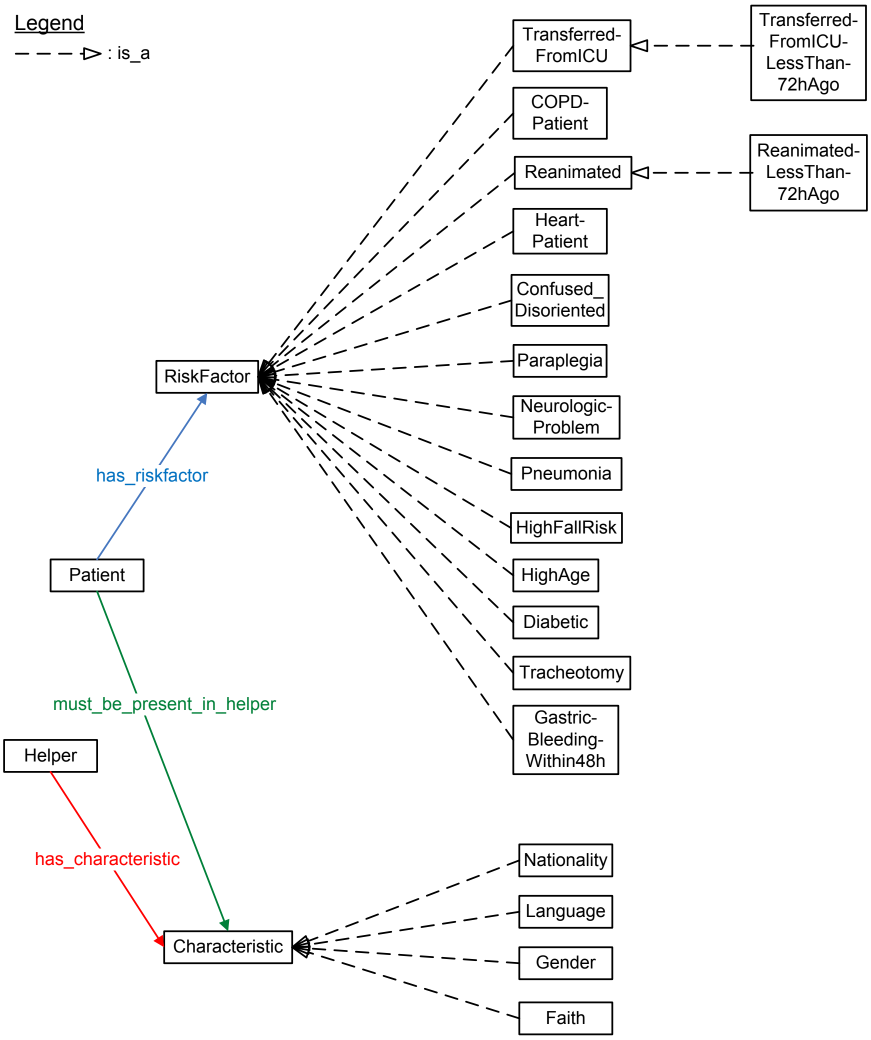


Figure 4.4: ***Fragment of the ontology that models the context information about the characteristics and risk factors.*** *Fragment of the ontology which models (1) the characteristics of the helpers and (2) the risk factors of the patients. To highlight the possibilities of the system, a (not exhaustive) list of risk factors was assembled by experts from both the medical and nurse call domain. The squares represent the classes. The arrows with the striped lines indicate subclass relationships. The other arrows and lines indicate relations between classes (object properties).*

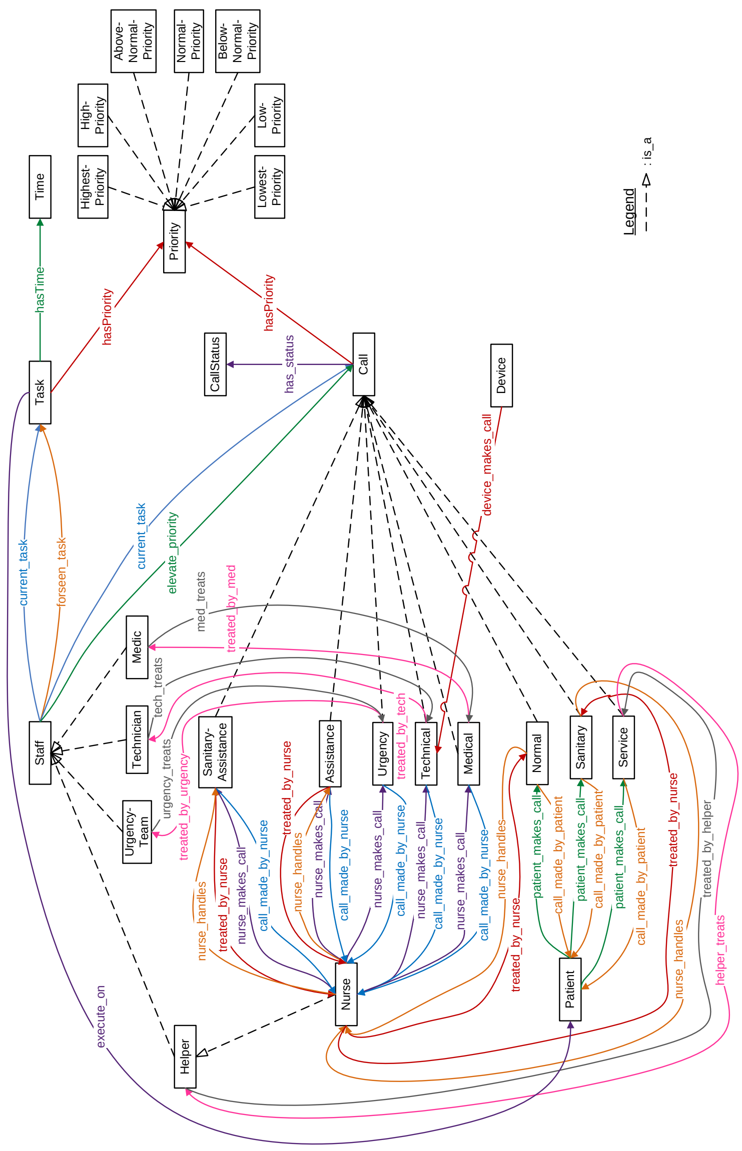


Figure 4.5: ***Fragment of the ontology that models the context information about the calls and tasks.*** *Fragment of the ontology which models the calls and tasks. It mainly indicates which calls can be made by patients and staff members and which staff members are allowed to handle these calls. Additionally, it models the possible priorities that a calls or tasks can have. The squares represent the classes. The arrows with the striped lines indicate subclass relationships. The other arrows and lines indicate relations between classes (object properties).*

4.2.2.2 Model of the calls and tasks

Each staff member has an associated current task, as can be seen in Figure 4.5. For each staff member, it is logged if this staff member is free or busy. Staff members can be handling a call or doing other tasks, e.g. giving medication to a patient. For each task the time by which the task should be completed and the patient for whom this task should be done can be indicated. It is also possible to maintain a list of tasks that a staff member should complete. A task can also be assigned a priority.

A general upper class maintains all the information that is applicable to each call such as the sequence number, the start and end time and the persons who made and handled the call. Each kind of call also has a time-out time. A call can have different statuses. When a call is launched, it has the status `Active`. This status changes to `Answered` when a staff member has been called. When the staff member is treating the call, the status changes to `Busy`. When the job is completely finished, the status is set to `Finished`.

The different specific calls that can be made are modeled as subclasses of this general upper `Call` class. For each call it is indicated which kind of person can make the call. As can be seen in Figure 4.5, three kinds of calls can be launched by patients. A normal call is made for medical problems and a service call is made for a "caring" task. When a normal call is made inside a sanitary room the call is automatically transformed to a sanitary call. All the other calls, namely urgency, medical, technical and (sanitary) assistance calls, are launched by nurses. Which kind of staff member can answer the call is also maintained.

The probabilistic assignment of patients to risk groups is used to determine the priority of the calls. There are seven classes of priorities: `Highest`, `High`, `Above Normal`, `Normal`, `Below Normal`, `Low` and `Lowest` priority as is illustrated in the upper right corner of Figure 4.5. The priority of a call is also based on its kind e.g. normal or sanitary. So when a patient from a risk group, makes a certain kind of call, this call is assigned a probability of having a certain priority. For example, when a high risk patient makes a normal call, this call has 2% chance of having a high priority. For now, these probabilities were determined by domain experts at Televic NV. The different devices that can be present inside a hospital also have to be taken into account. Devices such as heart monitors are able to launch technical calls when, for example, their cable is unplugged.

4.2.3 Algorithms

Several algorithms were constructed to assign the best possible nurse to a call. The first subsection details the algorithm that was used to reason with the probabilistic information to assign a more informed priority to a call that is based on the risk factors of a patient. The second subsection details the algorithm that was used to

Patient has risk factor	High Risk	Medium Risk	Low Risk
Diabetes	[0.5,1]	[0,0.3]	[0,0.2]
Heart disease	[0.5,1]	[0,0.4]	[0,0.1]

Table 4.1: The probabilistic assignment of patients to risk groups based on their risk factors

assign the most suitable nurse to a call.

4.2.3.1 Priority assessment of a call

The general probabilistic information in the ontology about the assignment of patients to risk groups and the priorities of calls can be used to determine the priority of a specific call made by a specific patient. For this the platform needs to reason about the general probabilistic information in the ontology and apply it to the situation at hand.

To model the probabilistic information in the ontology and reason about it, *Pronto* [26] was used. *Pronto* implements a probabilistic extension of *Description Logics (DLs)* [27], the First-Order Logic on which OWL is based [28]. *Pronto* was chosen because it is easy to use and understand and offers a wide range of reasoning support. All the reasoning is done in a totally logical way without an implicit or explicit translation of the *Knowledge Base* to for example a *Bayesian network*. By using *Pronto*, the probability that a specific call made by a specific patient has a certain priority can be determined. For example, suppose we have a patient, called `Patient1`, who has two risk factors, namely `Diabetes` and a `Heart disease`. `Patient1` then makes a `Normal` call. The ontology contains the probabilistic information (as probabilistic intervals) that a patient with one of these risk factors is a `High`, `Medium` and `Low Risk` patient, as can be seen in Table 4.1. *Pronto* reasons on this information to conclude that `Patient1` has [0.5,1], [0,0.3] and [0,0.1] chance of being a `High`, `Medium` and `Low Risk` patient respectively. The ontology also contains probabilistic information about the probability that a patient from a particular risk group makes a `Normal` call with a particular priority, as shown in Table 4.2. *Pronto* combines this information with the previously calculated probability intervals which indicate that `Patient1` is a `High`, `Medium` and `Low Risk` patient. *Pronto* concludes that the `Normal` call of `Patient1` has respectively [0,1], [0.1,0.6], [0.3,0.8], [0.1,0.6], [0,1], [0,1], [0,1] chance of having the `Highest`, `High`, `Above Normal`, `Normal`, `Below Normal`, `Low` and `Lowest` priority.

As shown in the previous example, *Pronto* calculates for each of the seven possible priorities, the probability that the call has this priority. However, one priority needs to be assigned to the call, so this priority can be used in the nurse call algorithm, see *The nurse call algorithm* Subsection of the *Algorithms* Section. To

Normal call made by	**Highest**	**High**	**Above Normal**	**Normal**	**Below Normal**	**Low**	**Lowest**
High risk patient		0.2	0.6	0.2			
Medium risk patient			0.3	0.6	0.1		
Low risk patient				0.6	0.3	0.1	

Table 4.2: The probabilistic assignment of calls to a priority category

resolve this issue, the following threshold algorithm was employed on the lower bound of the probabilistic intervals. If the probabilistic value for the highest priority class is higher than or equal to the threshold for the highest priority class, it gets the highest priority. If not, the same condition is checked for high, above normal, normal, below normal, low and lowest priority classes. The thresholds can be determined based on the specific characteristics, e.g. number of calls, needs and preferences of the department or hospital. The threshold that were used for the simulations are detailed in the *Collected data* Subsection of the *Evaluation set-up* Section. If the thresholds are 0.21, 0.3, 0.24, 0, 0.05, 0 and 0, ordered from the `Highest` to the `Lowest` priority, than the `Normal` call of `Patient1` from the previous example gets the `Above Normal` priority according to this threshold algorithm.

Although, the 0.2 release of *Pronto* increases the performance of the reasoning tasks over a single probabilistic statement, scalability is still a problem [29]. Currently *Pronto* can handle about 15 probabilistic statements in reasonable time. As a result, *Pronto* cannot currently handle all the probabilistic statements that were added to the ontology in reasonable time.

The following optimization was used in the *oNCS system* to speed up the probabilistic reasoning. First, during down-time, the probabilistic values that indicate that this patient is a high, medium or low risk patient are calculated and stored as known facts in the ontology. This does not have to be repeated often as risk factors do not change a lot during a patients stay in the hospital. Next, when a call is made, all the probabilistic statements that are needed to calculate the priority of this call are extracted from the ontology. Each time, at most 12 probabilistic statements will be extracted, namely the statements about the probabilistic assignment of this patient to the risk groups (3 statements) and the statements about the generic probabilistic assignment of this kind of call to the priority groups (9 statements).

4.2.3.2 The nurse call algorithm

A new algorithm was designed to find the correct staff member to handle a call. It uses the information stored in the ontology. It first determines which kind of calls has been made as can be seen in Figure 4.6.

Normal, sanitary, service and (sanitary) assistance calls employ the same basic algorithm which is visualized in Figure 4.7. The difference is that for normal,

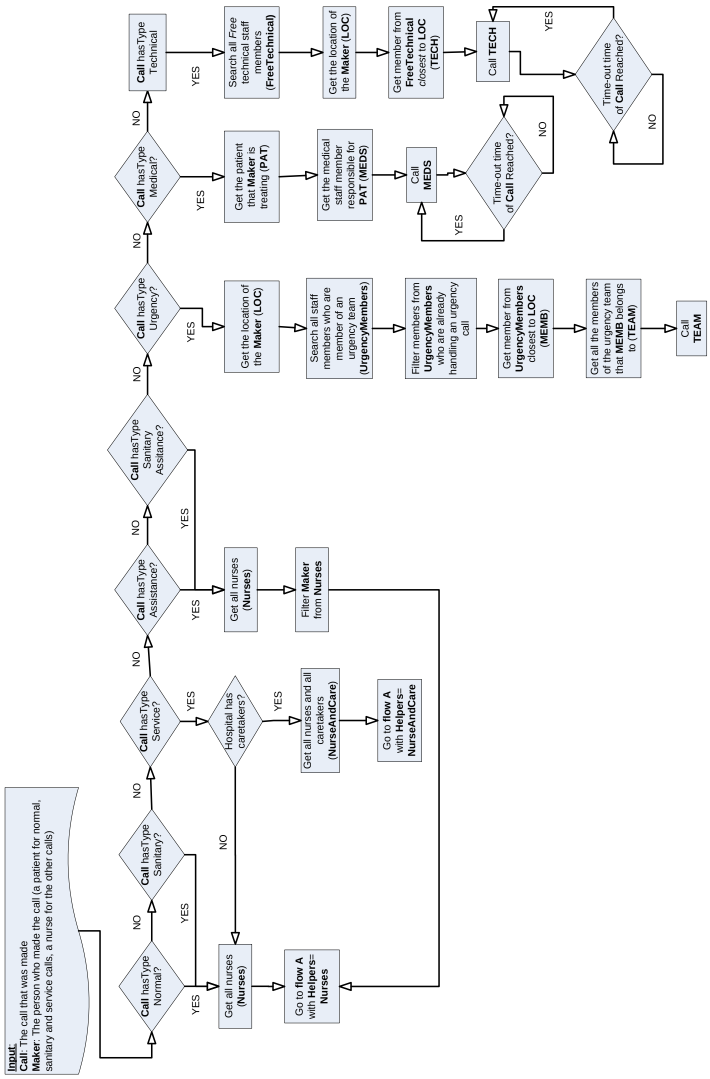


*Figure 4.6: **oNCS algorithm to find the correct staff member to handle a call.** This figure shows the flow chart of the oNCS algorithm, which finds a correct staff member to handle a call. It first determines which kind of calls has been made. Normal, sanitary, service and (sanitary) assistance calls employ the same basic algorithm which is visualized in Figure Figure 4.7 (Flow A). The difference is that for normal, sanitary and (sanitary) assistance calls only nurses can be called. For service calls caretakers can also be called. It is also made sure that the nurse that made the (sanitary) assistance call, cannot be called to answer this call. Urgency, medical and technical calls each have their own algorithm, which is visualized in this figure.*

sanitary and (sanitary) assistance calls only nurses can be called. For service calls caretakers can also be called. It is also made sure that the nurse that made the (sanitary) assistance call, cannot be called to answer this call.

The common algorithm first checks if the responsible nurse or caretaker can be called. Note that this responsible staff member can also be called if he/she is busy with a task that has a lower priority than the current call. If the responsible nurse or caretaker cannot be called, all the helpers who work on the department where the patient who the call is for lies are investigated. It is assumed that a nurse, who works on a department where the patient lies on, has more background information about the illnesses and concerns of this patient. Only for calls with the highest or high priority helpers are considered that are busy with a task with a lower priority. Otherwise these helpers will never be able to finish the work for the patients they are responsible for. If this option still does not offer a solution, the search is widened beyond the scope of the department and the helpers in the whole hospital are taken into account. If the result is empty again, this means that there are no available nurses in the direct vicinity. The distance becomes a deciding factor at this moment, so the closest nurse with right properties is selected, e.g. free, willing and qualified. If this still does not offer a solution, all the nurses in the hospital are considered and the one who is closest to the patient is called. Note that the characteristics are only used to choose among different available nurses. They are never used to decide that a nurse cannot handle a patient.

The algorithm has a time-out procedure. If a staff member has not indicated that he/she is going to handle the call within the time-out time that is specified for this type of call in the ontology, another staff member is selected to handle the call by running the algorithm again.

Urgency, medical and technical calls each have their own algorithm as can be seen in Figure 4.6. For urgency calls, the priority lies on finding a person who is near instead of a person who is free. This is necessary because lives are at stake when an urgency call is issued. A time-out procedure is not needed here, as an urgency call will always be immediately answered. The algorithms for the technical and medical calls are rather simple and straightforward because they generally have a very low priority.

Note that a staff member can sometimes be called while he/she is already busy with a task. It is up to the staff member to decide if he/she is going to interrupt his/her current task or not. In contradiction to the place-oriented case, the staff member knows that the new call has a higher priority than the task that this staff member is currently working on. Based on these priorities the staff members can make a more funded decision to interrupt their current task or not. If the staff member decides to answer the new call, the system automatically interrupts the current task of this staff member. If the task is a call, another staff member is searched to handle the call. If it is not a call, the task is added to the list of tasks

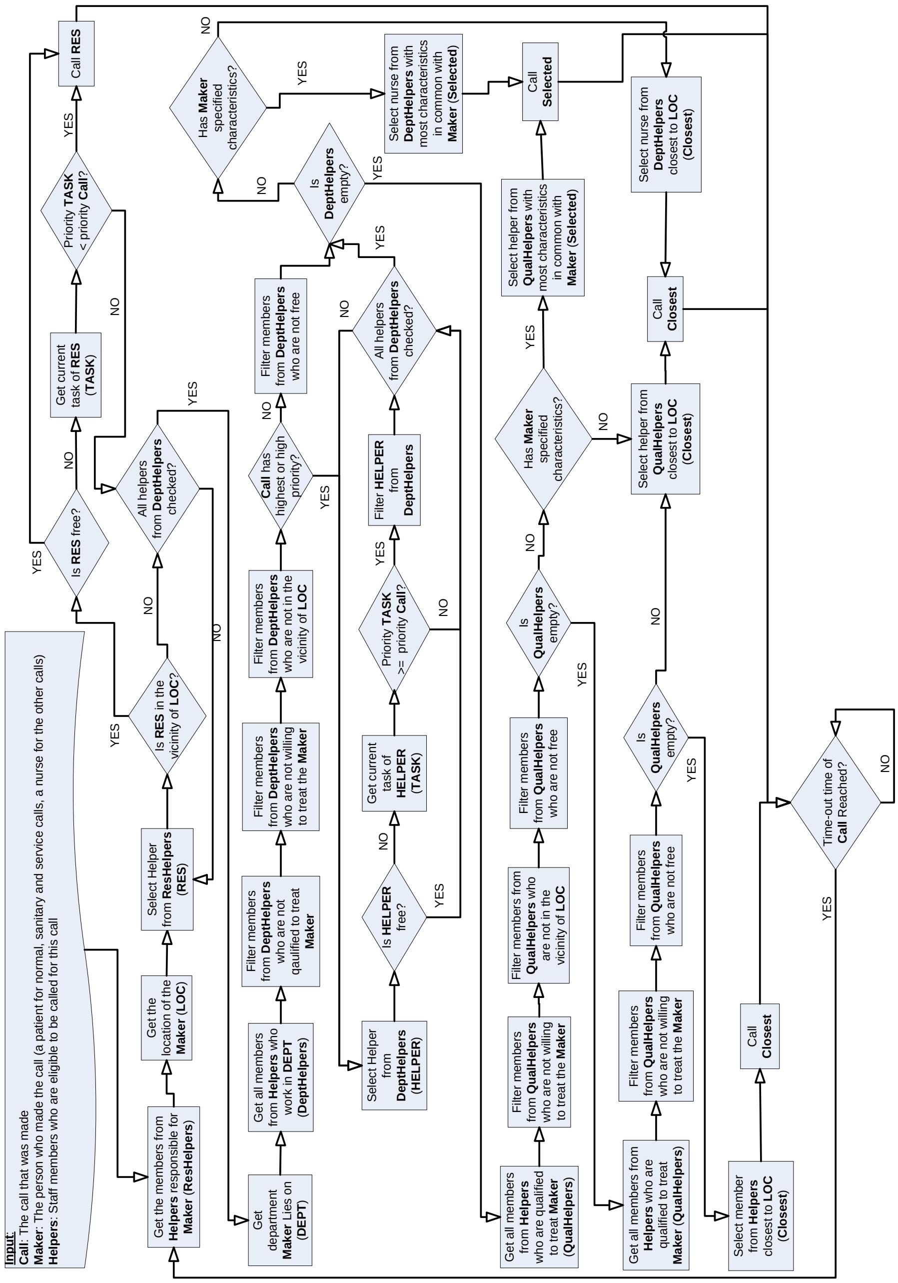


Figure 4.7: ***Flow A: oNCS algorithm to find the correct staff member to handle a normal, sanitary, service or (sanitary) assistance call.***

that this staff member must do. So the staff member does not have to remember himself that he/she has to return to a task or that he/she has to call some other staff member.

4.2.4 Implementation details

This section gives an overview of the implementation of the entire *oNCS system*. The first Subsection, *Building the ontology*, details how the ontology was digitalized. The second Subsection, *the oNCS platform*, details how the algorithms were implemented and were integrated into the existing *Context-Aware Service Platform (CASP)*.

4.2.4.1 Building the ontology

Different languages exist to digitalize an ontology. The *Ontology Web Language (OWL)* [30] was chosen for a number of reasons. First, *OWL* is a recommendation by the World Wide Web Consortium (W3C) [31] and is the most widely used and well-known ontology language. Secondly, using one of the three sublanguage flavors of *OWL*, *OWL-Lite*, *OWL-DL* and *OWL-Full*, one can easily adapt to the required expressiveness at hand. *OWL-DL* is based on *Description Logics* [27], a decidable part of *First Order Logic*. This ensures that reasoning on *OWL-DL* models is computationally complete and decidable, which means that all computations will end in finite time. Thirdly, there also exist a wide range of tools for *OWL* such as editors and visualization tools. Sophisticated *Reasoners* exist that allow checking the consistency and classifying the ontology. *OWL* can also easily be integrated with different *Rule* platforms and can be queried with *SPARQL* [32]. Moreover, *OWL* is the only ontology language for which there exist mature tools to express and reason about probabilistic knowledge. A final advantage is the straight forward integration of an *OWL* ontology into the *CASP framework*, see the *oNCS platform* Subsection, by using *Jena* [33], a *Java* framework for building *Semantic Web* applications.

The *Protégé* editor [34] was used to develop the deterministic part of the ontology. The *Pellet Reasoner* [35] was used to check the consistency and the classification of the ontology. To use the probabilistic *Reasoner Pronto*, probabilistic statements have to be expressed in an *OWL*-file by using axiom annotations, which is a new feature of *OWL 1.1* [36]. As *Pronto* supports probability intervals, the intervals specified in the *Profile management* Subsection can be used. The exact probabilities were expressed by axioms that were annotated with probability intervals with an equal upper and lower limit.

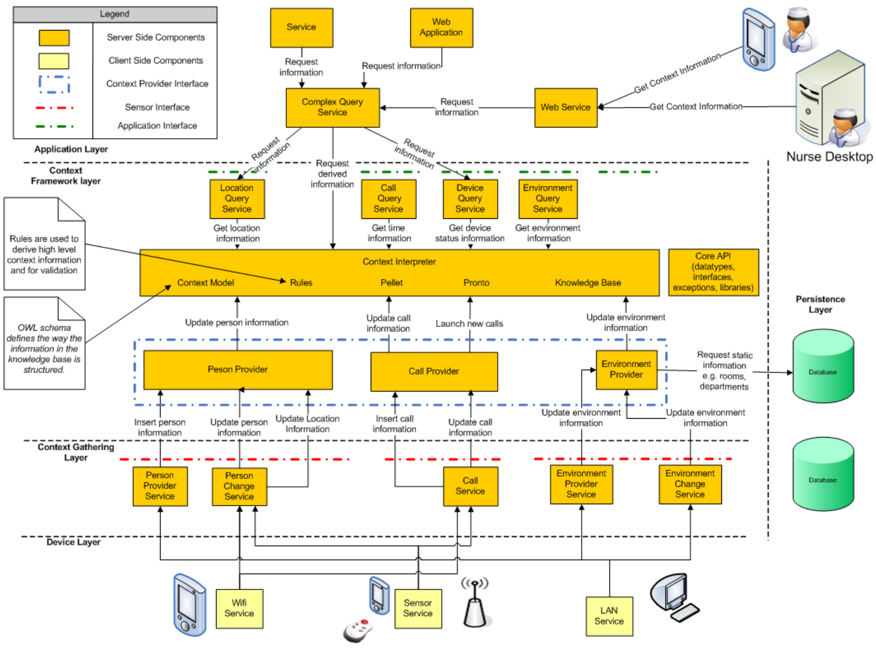


Figure 4.8: ***The architecture of the oNCS platform.*** *This figure represents the architecture of the oNCS platform. The Context Framework Layer is the most important layer. Within this layer the Context Interpreter controls all the context information. The ontology determines the structure of the Knowledge Base. The Knowledge Base contains all the data that conforms to the ontology. The Context Model provides access to the ontology by using Jena. Pellet is used to check the consistency of the model. The layer also holds all the Rules that work with the information in the Knowledge Base. The different Context Providers allow importing external information into the framework. This information is then added to the Knowledge Base. This new information can come from a database (Persistence Layer) or directly from a device (Device Layer and Context Gathering Layer). Currently three Context Providers are provided: the* `Person Provider`*, the* `Environment Provider` *and the* `Call Provider`*. The Query Services are used to extract information from the Knowledge Base. The Query Services can be used to visualize the knowledge or to use the information in another application (Application Layer). The methods in the Context Providers and Query Services were also made available as Web Services.*

4.2.4.2 The oNCS platform

The *oNCS platform* was built as an extension of the *Context-Aware Service Platform (CASP)* [18]. The *CASP framework* is a collection of bundles for *OSGi* that were developed to handle context information. The *OSGi Framework* [37] is an open service platform for the delivery and control of different applications and services to a certain type of networked device in the environment. In this case the devices would be the portable nurse call buttons, the sensor nodes, the PDAs and the nurse desktop. *OSGi* can best be seen as an application, which is called a bundle in *OSGi*, container. It is possible to plug new bundles into the *OSGi framework* at any time. This expands the framework with new possibilities and services. These new services can be dynamically discovered by the other bundles. So basically, *OSGi* technology provides the standardized primitives that allow applications to be constructed from small, reusable and collaborative components. The open source implementation *Knopflerfish* was used.

An overview of the *oNCS platform* is shown in Figure 4.8. The *Context Framework Layer* is the most important layer. Within this layer the *Context Interpreter* controls all the context information. The ontology determines the structure of the *Knowledge Base*. The *Knowledge Base* contains all the data that conforms to the ontology. The *Context Model* provides access to the ontology by using *Jena*. *Pellet* is used to check the consistency of the model. The layer also holds all the *Rules* that work with the information in the *Knowledge Base*.

The different *Context Providers* allow importing external information into the framework. This information is then added to the *Knowledge Base*. For example, the *Person Provider* is used by the sensor nodes to insert new information about the location of the patients and staff members. This new information can come from a database (*Persistence Layer*) or directly from a device (*Device Layer* and *Context Gathering Layer*). Currently three *Context Providers* are provided: the `Person Provider`, the `Environment Provider` and the `Call Provider`. All the *Context Providers* implement a common interface, namely `ContextProvider`, which makes it easy to plug new *Context Providers* into the framework.

The *Query Services* are used to extract information from the *Knowledge Base*. This ensures that application developers do not have to write the error-prone queries themselves. They also do not have to translate the results of the queries to usable *Java*-objects. The *Query Services* can be used to visualize the knowledge or to use the information in another application (*Application Layer*).

To make the platform more generic some *Web Services* were developed. These *Web Services* allow applications and devices from anywhere in the network to call methods to add new information to the *Knowledge Base*, such as making new patients, nurses or calls, or extract information, such as which nurse has been called to answer a call. These methods call the *Context Providers* and *Query Services* to add or extract the knowledge.

Note that the framework is modularly divided into bundles. These bundles can be plugged into the *Knopflerfish (OSGi)* framework and can dynamically discover each other. This also allows deploying the framework in a distributed manner, which is important when high performance is needed. The *oNCS platform* runs on multiple servers to ensure reliability and scalability. When a server goes down, another server can still process all the requests. Standard load-balancing algorithms [38, 39] can also be used to distribute the requests amongst the different servers.

To improve the scalability of the system, information that is no longer needed in the ontology can be stored in a database so it can be used for studies or analysis. This can for example be done at night. A lot of information can be removed from the ontology each day such as calls that have been completely handled or patients that have left the hospital. The server additionally also logs all the actions of the systems such as who added which information to the ontology, which calls were launched and who handled them.

The *oNCS* nurse call algorithm is implemented by using *Rules*. The *Rules* are activated when an event occurs in the *Knowledge Base* for example when a new call is added. When the condition is fulfilled, the *Rule* calls a *functor*. A *functor* does some calculations with the parameters it receives from the *Rule*, for example the new call. The *functor* can also change the information in the *Knowledge Base*.

Every kind of call that can occur is handled by a different *Rule*. For example, the following code fragment shows the *Rule* that reacts to a normal call:

```
[insert_nurse_normalcall:
(?x rdf:type ncs:Normal)
(?x ncs:has_status ?CallStatus)
(?CallStatus ncs:Kind 'Active')
noValue (?x ncs:treated_by_nurse)
→ findHelper (?x)]
```

As can be seen, this *Rule* is activated when a normal call is launched (its status is `Active` and no staff member has been called). If the condition is fulfilled the *functor* `findHelper()` is called which takes the call as argument. The *functor* follows the earlier stated algorithm specified in *The nurse call algorithm* Subsection of the *Algorithms* Section to find a correct staff member to handle the call. It adds the information that this particular staff member has to handle this particular call to *Knowledge Base* (the `treated_by` relation in the ontology). This guarantees that the *Rule* is not fired again, because the `noValue` condition is no longer fulfilled. All the other types of calls are handled in a similar matter.

Rules were also constructed that trigger when the status of a call is changed. The *Rules* adapt the *Knowledge Base* for example to indicate that a nurse is busy with a call, has finished a call, the time at which the call was finished and so on. Most importantly these *Rules* also automatically interrupt the current task (if any) of the called staff member as explained in *The nurse call algorithm* Subsection of the *Algorithms* Section.

A last set of *Rules* is used to implement the time-out procedure for each kind of call.

Note, that if a different nurse call algorithm should be used, e.g. because another hospital might use a different nurse call policy, only the *functor* needs to be rewritten. This can be easily done as a lot of re-usable methods and code have been provided e.g. to collect the needed information from the ontology, compare the preferences of the patient with the characteristics of the staff members or find the closest staff member.

4.2.5 Evaluation set-up

To test and demonstrate the advantages of the new *oNCS platform*, simulations were set up with realistic data about a nursing department of the Ghent University Hospital [19].

4.2.5.1 Collected data

The studied department of the Ghent University Hospital contains patients that are fairly mobile. They are not confined to their beds, but they do spend most of their time in their room. The most important mobile activities are going to the restaurant, going outside to smoke, getting the newspaper and being moved to other departments to undergo some additional medical examinations. The floor plan can be seen in Figure 4.9. The most important spaces to notice on the floor plan are the rooms and the sanitary areas. The department contains 26 beds and has an occupation rate of 84.62%. Each room has one or 2 beds.

The three most visited spaces by patients were included in the simulations, namely the smoking area just outside the building, the CT scanner and the cafeteria at the ground-level of the building. The time it takes to travel to all these spaces from the department was measured. Finally, it was determined how patients divide their time over these different spaces.

Some information about the staff in this department was also gathered. The department has three shifts: the early, late and night shift. During the week there are 5 nurses during the early shift, 4 during the late shift and 1 or 2 during the night shift. During the weekend there are 4 nurses during the early shift, 3 during the late shift and 1 or 2 during the night shift. The department also has a head nurse, but this head nurse never answers calls. Each nurse is responsible for approximately 5 or 6 patients during a shift. They are assigned based on the split-up of the rooms of the department according to the number of present nurses. Patients in adjacent rooms are assigned to the same nurse. A patient is never assigned to more than one nurse at the same time.

The walking behavior of the staff members was simulated by using information, which was gathered during an earlier study [40], about their tasks and the percentage of time they spend on each group of tasks, as visualized in Figure 4.10. For each of the tasks it was also determined if the task was always (low or lowest

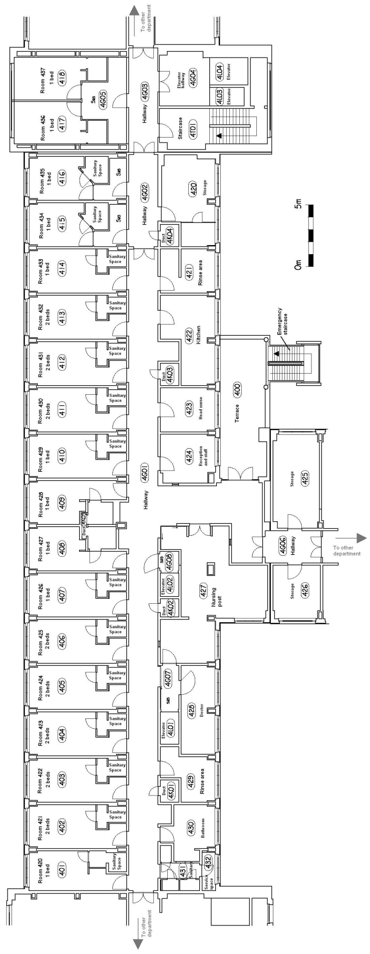


Figure 4.9: ***The floor plan of the studied department.*** *This figure represents the floor plan of the studied department of the Ghent University Hospital. The department contains patients that are fairly mobile. The most important spaces to notice on the floor plan are the rooms and the sanitary areas. The department contains 26 beds. The floor plan indicates for each room how many beds it contains. Most rooms have their own sanitary space, but there are also some shared sanitary spaces. The nursing post is the place where nurses reside when they are not helping patients. This space is used to for example prepare medication or write reports. The head nurse has her own office. Patients do not have access to the storage and service spaces, the terrace, the rinse areas and the kitchen. The doors on the left and right of the floor plan are used to go to other departments. Generally patients use the elevator on the right of the floor plan to leave the department. The elevator in the middle of the floor plan is generally only used by staff members.*

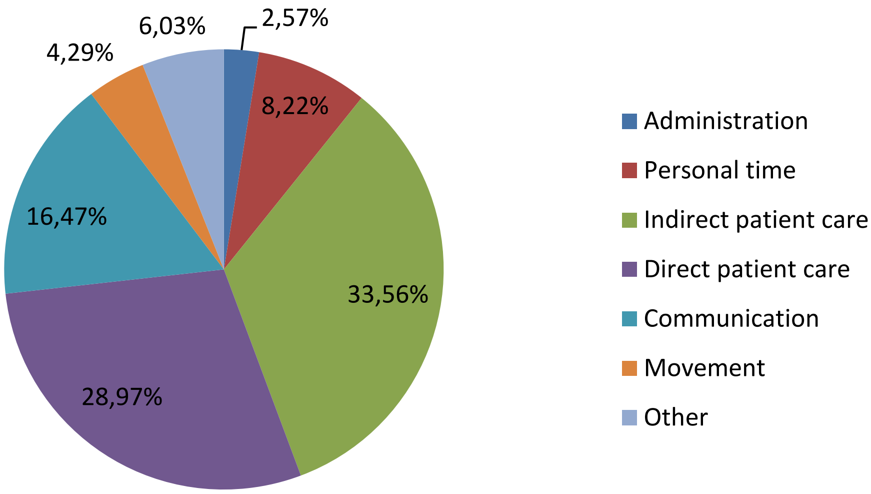


Figure 4.10: ***Distribution of time of the nurses across different kinds of tasks.***

priority), never (high priority) or sometimes interruptible (below, above or normal priority).

To monitor the added value of keeping the characteristics in the ontology, information was gathered about the spoken languages by both the patients and the staff members. All the staff members are able to speak Dutch, 80% of the staff members speak English, 70% speak French, 20% speak German and none of them speak Italian or Spanish. On the other hand, 2% of the patients only speak French and 3% of them only speak German.

It was determined how many patients have none, 1, 2 or more risk factors and which risk factors were more frequent than other risk factors by assigning a weight to them, as can be seen in Table 4.3. Some combinations of risk factors were deemed to be more frequent than others:

- COPD and tracheotomy
- High age and disoriented/confused
- High age and high fall risk
- Diabetes and disoriented/confused
- Neurological problem and disoriented/confused
- Transferred from the ICU and tracheotomy

Based on this data, patients were assigned risk factors.

Number of patients with:	
0 risk factors	10
1 risk factor	10
2 risk factors	8
$>$ 2 risk factors	2
Risk factor weights (%):	
High age (a)	50
Diabetes (b)	10
Heart disease (c)	3
High fall risk (d)	5
Neurologic problem (e)	3
Tracheotomy (f)	10
COPD (g)	3
Paraplegia (h)	3
Pneumonia (i)	3
Disoriented/ confused (j)	5
Gastric Bleeding within 48h (k)	3
Transferred from ICU (l)	1
Transferred from ICU within 72h (m)	0
Reanimated (n)	1
Reanimated within 72h (o)	0

Table 4.3: The distribution of the risk factors amongst patients in the three departments

The thresholds for the probabilistic reasoning algorithm, see the *Priority Assessment of a call* Subsection of the *Algorithms* Section, were determined by generating 22500 realistic calls and determining the priority each call gets by adjusting the threshold. The combination of thresholds was searched for which the percentages of calls assigned to a certain priority deviated least from the following distribution: 5% calls with highest priority, 10% with high priority, 25% with above normal priority, 35% with normal priority, 25% with below normal priority and 0% with the low and lowest priority. This distribution reflects a realistic hospital environment. The tested kinds of calls generally do not have the low or lowest priority as these categories are preserved for medical and technical calls. The middle categories, namely above normal, normal and below normal, generally contain more calls as most calls are made for simple requests. The chosen thresholds are 0.21 for the highest priority, 0.3 for the high priority, 0.24 for the above normal priority, 0 for the normal priority, 0.05 for the below normal priority and 0 for the low and lowest priority.

4.2.5.2 Current nurse call algorithm

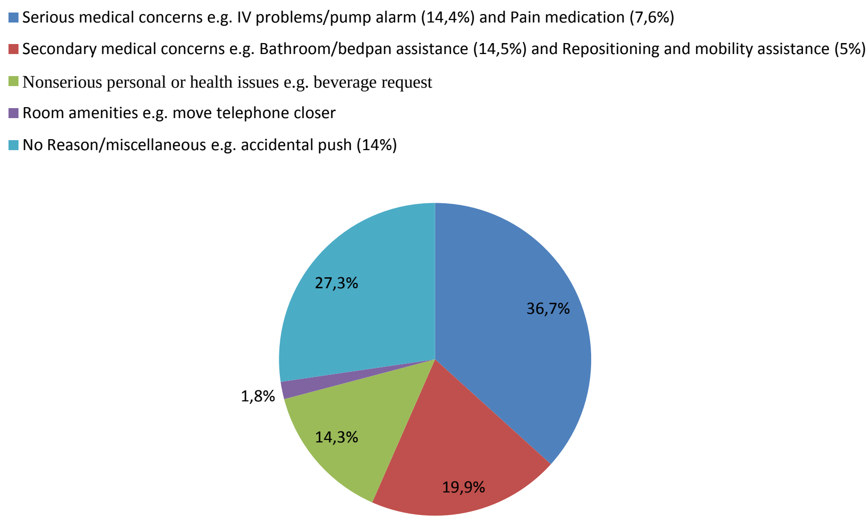


Figure 4.11: ***Reasons for patients' call light use [8].***

When calls are made by patients inside rooms, they are treated as normal calls. When calls are made inside a sanitary space, they are treated as sanitary calls. Nurses are able to make (sanitary) assistance calls in this department, but there are no buttons to make urgency or medical calls. Technical calls are not taken into account in the simulations as the result is straight-forward. Technical calls always get the lowest priority and a member of the technical staff is called as explained in *The nurse call algorithm* Subsection of the *Algorithms* Section.

Information about the calls, such as frequency and duration, was gathered during three weeks by studying the logging information of the currently installed place-oriented nurse call system. Limited research has been done on reasons for patients' call light use in the Ghent University hospital. The paper by Meade [8] presents an extensive study about this subject. The used results are presented in Figure 4.11. When a call is made a reason is randomly assigned based on these percentages. The average time that a nurse spends on handling a task from each category was also determined in an earlier study [40].

The normal, sanitary and (sanitary) assistance calls are handled as follows. All the nurses of the department receive the calls on their beepers or portable phones. A light also switches on above the room of the patient. The nurse, who arrives first at the location of the patient, switches off the call and starts treating the patient. If the time-out of a call is reached and none of the staff members have come to handle the call, all the nurses of the department receive the call again. As can be seen, it is possible that multiple nurses arrive at a room to handle a call, as multiple nurses are called and one nurse does not know if another nurse will handle the call or not. If they interrupt their current task (which could also be a call) to handle this call, the nurses have to remember themselves that they have to go back to that interrupted task. In case of an interrupted call, the other patient also has to wait until the nurse has finished this call, while it could be of a lower priority.

4.2.5.3 Simulation set-up

A realistic day-to-day hospital scenario was simulated. This means that the beds in the department are occupied averaging around the occupation rate as indicated in the *Collected data* Subsection. Of course it is assumed that the patients already own portable buttons and can thus move around freely and still make calls. When this situation is simulated for the place-oriented system, some calls may be impossible to handle e.g. calls made in the middle of a hallway. The movements of the patients were determined out of the collected data about the mobility of the patients and their tendency to visit other areas. During these movements they can make (sanitary) calls modeled according to a Poisson process with λ=0.001164021. Once a patient makes a call, it is assumed that this patient stands still. The movements of the nurses were determined out of the collected data about how they divide their time around their different kinds of tasks. During these tasks they receive calls of patients. They will only interrupt their current task, if the call has a higher or equal priority. They will only interrupt current calls, if the new call has a higher priority. If the new call does not have a priority, as can occur in the place-oriented system, a nurse chooses randomly to interrupt his or her current task or call. During the handling of a call, nurses will launch a (sanitary) assistance call with a probability of 0.07386%. If a nurse has to choose between multiple calls to handle, it is assumed that the nurse chooses the one with the highest priority. If the calls do not have priorities or multiple calls have the same priority, the closest call is chosen. It is assumed that patients or nurses that are on the move advance 1 meter in the direction of their goal during each time step. Characteristics of patients and nurses, risk factors of the patients and responsibility of staff members for certain patients were simulated as indicated in the *Collected data* Subsection.

The simulation was done 30 times for each of the 3 shifts during the weekend and 30 times for each of the 3 shifts during the week. These simulations were done on a PC with the following specifications: Intel Core 2 Duo Processor P8600 (2.40GHz, 1066MHz, 3MB), 4 gigabyte of RAM (2 x 2 gigabyte) and a 250 GB Serial ATA (7200RPM) hard drive.

4.3 Results

Both the *oNCS* and the place-oriented system were simulated in a realistic hospital setting. The first subsection details the results of the comparison between the two. The advantages of the probabilistic risk assessment algorithm were also evaluated. Finally, the performance of the system is discussed.

4.3.1 Simulation Results

As mentioned in the *Current nurse call algorithm* Subsection of the *Evaluation set-up* Section, it is possible that multiple nurses arrive at a room to handle a call in the place-oriented system. On average 0.43 unnecessary nurses arrived at a call per simulation, with a maximum of 4 nurses in 1 simulation. This means that, at least one nurse each day arrives at a call which is already being treated by another nurse.

As mentioned in the *Simulation set-up* Subsection of the *Evaluation set-up* Section, some calls are impossible to handle in the place-oriented system as they are made in a place, e.g. the hallway, where currently no buttons are provided. These can be handled by the *oNCS system* as the patients have portable buttons. On average 2.53 impossible calls were made per simulation, with a maximum of 12 impossible calls in 1 simulation. This means that each shift about 3 calls cannot be handled in the current system. Especially the worst case scenario with 12 impossible calls is alarming. Patients can walk around in the hallways and staircases and are unable to make calls. Especially outside in the smoking area there are no staff members close, who could help the patient fast.

Figure 4.12 shows the number of calls that have a nurse present as a function of the arrival times of these nurses. This means that the nurse has arrived at the place where the patient made the call. Note that the first part of the x-axis has a

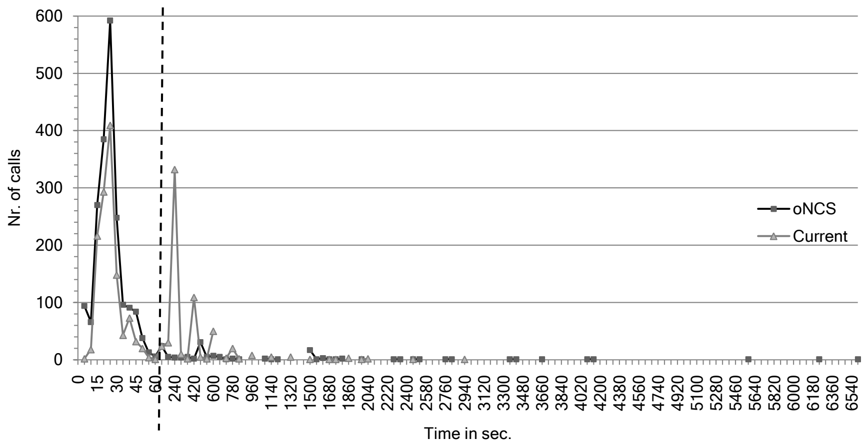


Figure 4.12: ***Number of calls as a function of the nurse arrival times.*** *This figure shows the number of calls that have a nurse present (y-axis) as function of the arrival times of these nurses (x-axis in seconds) for both the oNCS system and current, place-oriented system. This means that the nurse has arrived at the place where the patient made the call. Note that the first part of the x-axis has a time-step of 5 seconds, while the second part has a time-step of 60 seconds. The two parts are separated by the striped vertical line.*

time-step of 5 seconds, while the second part has a time-step of 60 seconds. Most of the calls have a nurse present after 60 seconds in the *oNCS system*. In the place-oriented system about half of the calls have a nurse present after 60 seconds. Most of the rest of the calls are handled after 780 seconds.

The difference can be easily explained. In the *oNCS system* only one nurse receives the call. In most cases the call will have a higher priority than the current task of the nurse because the algorithm takes this into account. Therefore, the nurse will immediately go and answer the call. In most cases the distance to the patient will be limited, as this is taken into account in the novel nurse call algorithm.

On the other hand in the place-oriented system, multiple nurses receive the call. They have to decide if they are going to quit their current task. They have to make this decision without information about the call. So in the case that all nurses ignore the call, thinking someone else will handle it, the call has to be relaunched before it is noticed that nobody went to handle the call. This is illustrated nicely on the graph, as a peak can be seen each time the call is relaunched, namely shortly after 180, 360, 540, seconds. Moreover, the distance is not taken into account when calling the nurses in the place-oriented system. So it is possible that the nurse must walk a long time before arriving at the room of the patient.

The tail of the *oNCS system* is much longer than the place-oriented system. This is caused by the impossible calls which are not answered in the place-oriented system, but which are answered in the *oNCS system*. Most of these calls occur in places that are very far away from the department e.g. normal calls in the smoking area and restaurant or assistance calls in the scanner room. This could be solved by allowing nurses from closer departments to answer these calls. However, these nurses were not included in the simulations.

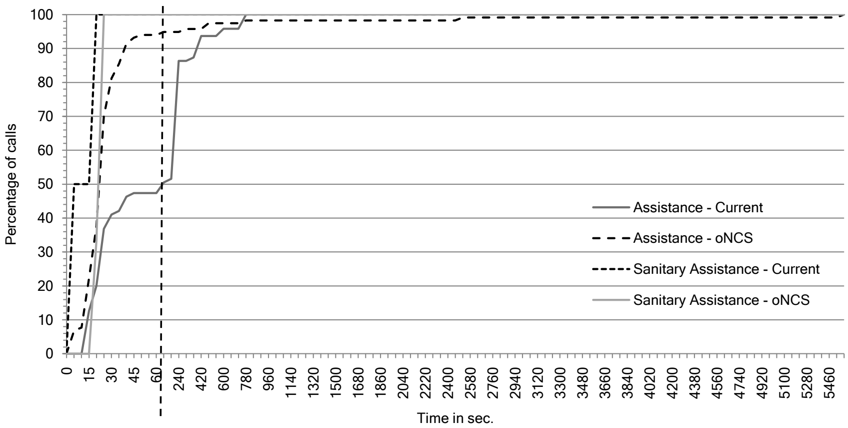


Figure 4.13: ***Percentage of (sanitary) assistance calls as a function of the nurse arrival times.*** *This figure shows the percentage of assistance and sanitary assistance calls (y-axis) as function of the arrival times of these nurses (x-axis in seconds) for both the oNCS system and the current, place-oriented system. This means that the nurse has arrived at the place where the patient made the (sanitary) assistance call. Note that the first part of the x-axis has a time-step of 5 seconds, while the second part has a time-step of 60 seconds. The two parts are separated by the striped vertical line.*

As can be seen in Figure 4.13, 100% of the sanitary assistance calls have a nurse present within 15 seconds in both the current and *oNCS system*. This is because these calls generally have a very high priority. The *oNCS system* is slightly slower than the place-oriented system which can be explained by the initial delay of calling the nurse call algorithm (see the *platform performance* Subsection). However, for the assistance calls the *oNCS system* performs much better than the place-oriented system. In the oNCS system 95% of the assistance calls have a nurse present within the first minute. In the place-oriented system this only occurs after 480 seconds (8 minutes).

A similar scenario can be spotted for the sanitary calls in Figure 4.14. In the *oNCS system*, 100% of sanitary calls have a nurse present after 40 seconds. In the place-oriented system, only 72% of the sanitary calls are handled at this point and takes 960 seconds (16 minutes) until all the sanitary calls have a nurse present. The normal calls generally also have a nurse present faster in the *oNCS system*. 90% of these calls have nurse present within 45 seconds. In the place-oriented system only

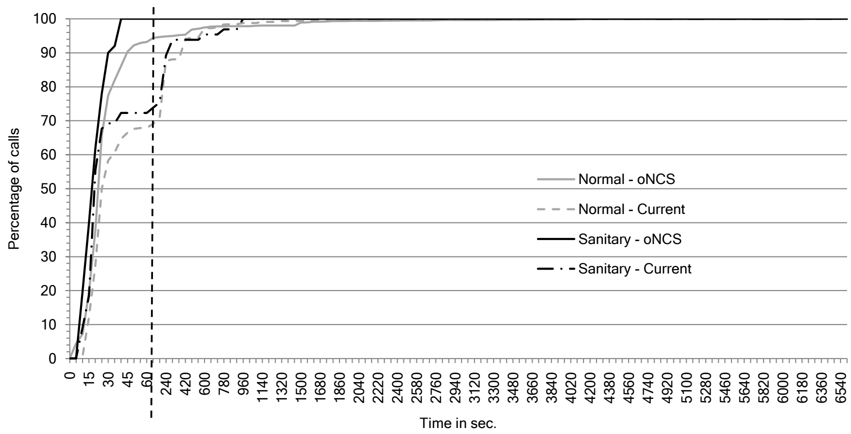


Figure 4.14: ***Percentage of normal and sanitary calls as a function of the nurse arrival times.*** *This figure shows the percentage of normal and sanitary calls (y-axis) as function of the arrival times of these nurses (x-axis in seconds) for both the oNCS system and the current, place-oriented system. This means that the nurse has arrived at the place where the patient made the normal or sanitary call. Note that the first part of the x-axis has a time-step of 5 seconds, while the second part has a time-step of 60 seconds. The two parts are separated by the striped vertical line.*

66% of the calls have a nurse present then. It reaches 90% after 300 seconds (5 minutes). A small percentage of these calls take a long time to be handled, notably for the assistance calls and normal calls in the *oNCS system*. This can again be explained by the impossible calls which are answered in the *oNCS system*, but not in the place-oriented system.

The number of calls that have a nurse present as a function of the arrival times of these nurses for different call priorities are visualized in Figure 4.15 for the *oNCS system* and in Figure 4.16 for the place-oriented system. As can be seen the distribution of the calls amongst the different priority levels is as to be expected. The below normal priority is assigned the most. This department contains a considerable amount of patients without any risk factors, when they make a normal call it will get the below normal priority. Moreover some patients with a minor risk factor would also make normal calls that get this priority. The normal and above normal priorities are assigned to a comparable amount of calls. These are for example sanitary calls or calls made by patients with some risk factors. The highest priority gets assigned to the least amount of calls. These are primarily sanitary assistance calls or assistance calls made by patients with some risk factors.

The amount of time it takes for a nurse to be present after the call is made in the *oNCS system* is also as expected. Calls with the highest priority are handled the fastest. Most of those calls have a nurse at the scene within 45 seconds. The

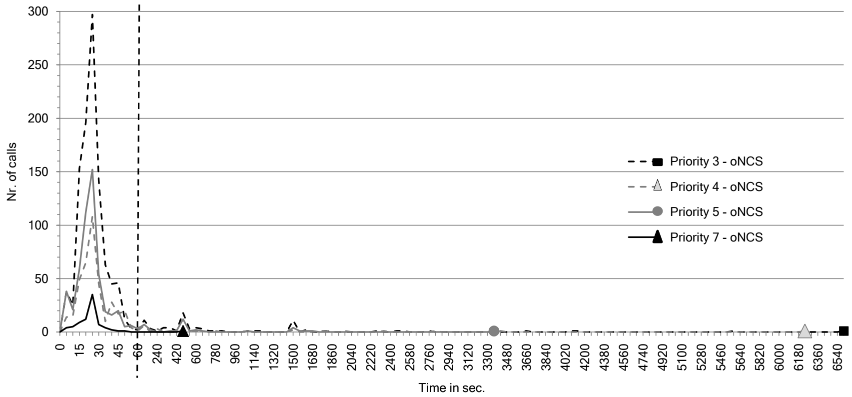


*Figure 4.15: **oNCS system: number of calls as function of nurse arrival times for different call priorities.** This figure visualizes the number of calls that have a nurse present (y-axis) as function of the arrival times of these nurses (x-axis in seconds) for different call priorities for the oNCS system. This means that the nurse has arrived at the place where the patient made the call. This allows evaluating (1) the influence of the priority of the call on the arrival time of the nurse (2) the distribution of the calls amongst the different priorities. Note that the first part of the x-axis has a time-step of 5 seconds, while the second part has a time-step of 60 seconds. The two parts are separated by the striped vertical line.*

worst case scenario still has a nurse at the scene within 480 seconds. The calls with below normal, normal and above normal priorities are handled somewhat slower but most calls still have a nurse in place within 60 seconds. The tails are longer, but still in the correct order: the worst-case time of the calls with above normal priority is lower than the worst-case time of calls with the normal priority which is in turn lower than the worst-case time of the calls with the below normal priority.

However, the amount of time for a nurse to be place is not as logical in the place-oriented system. It is obvious that the place-oriented system does not take the priority of the call into account. The different peaks can be explained by the relaunch times of the calls. Every 180 seconds a call which does not have a nurse in place is relaunched. It can be seen that after these time points (0, 180, 360,) a series of calls is answered. Even calls with the highest priority need to be relaunched up to 4 times before someone is in place. The calls with the above normal, normal and below normal priority have the same trend of having a nurse in place within a certain time. Only calls with the above normal priority seem to be handled faster than the other calls. However, the calls with this priority also have the longest worst-case time.

Table 4.4 gives an overview of the distribution of calls amongst the nurses present in the department. The first column indicates the number of nurses that

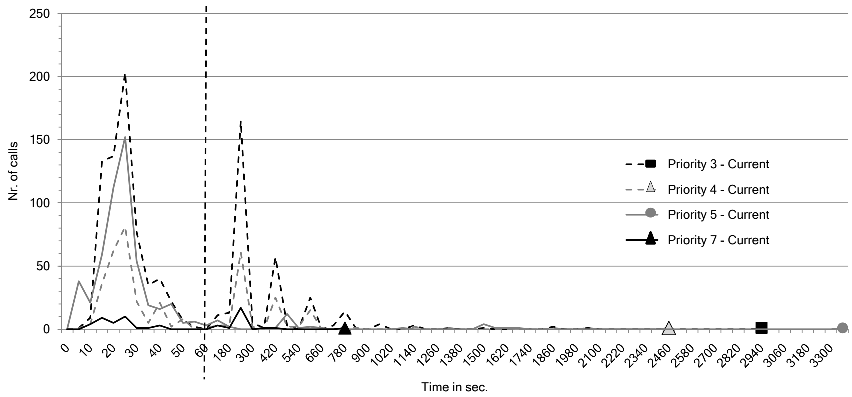


Figure 4.16: ***Place-oriented system: number of calls as function of nurse arrival times for different call priorities.*** *This figure visualizes the number of calls that have a nurse present (y-axis) as function of the arrival times of these nurses (x-axis in seconds) for different call priorities for current, place-oriented system. This means that the nurse has arrived at the place where the patient made the call. This allows evaluating (1) the influence of the priority of the call on the arrival time of the nurse (2) the distribution of the calls amongst the different priorities. Note that the first part of the x-axis has a time-step of 5 seconds, while the second part has a time-step of 60 seconds. The two parts are separated by the striped vertical line.*

were present in the department during the simulation. For both systems it is shown what the maximum and minimum percentage of calls was that a nurse handles during a shift. It is also indicated how many nurses handle zero calls during a shift. Finally, the standard deviation is given between the percentage of calls that nurses handles and the mean. The mean is of course the ideal percentage of calls that a nurse should handle, for example 50% in the case there are 2 nurses in the department. Note, that the number of calls that a nurse handles per shift is different from the amount of calls that this nurse is assigned and thus receives on the portable phone or beeper.

The *oNCS system* leads to a slightly better workload distribution than the place-oriented system. Especially in the case when there are only 2 nurses available in the department all the calls are divided more evenly amongst the different nurses. In the *oNCS system*, all the nurses get about 50% of the calls, while in the place-oriented system some nurses get up to 70% of the calls (while the other nurse in the department at that time thus only gets 30% of the calls). The difference is also obvious in the case of four and five nurses as the extremes are made less extreme in the *oNCS system*.

Nr. of nurses in the department	Workload distribution Place-oriented system: Max.	Min.	# 0%	Std. Err.	Workload distribution *oNCS system:* Max.	Min.	# 0%	Std. Err.
1	100	100	0	0	100	100	0	0
2	70.97	29.03	0	11.90	57.14	42.86	0	5.12
3	58.33	12.12	0	12.65	60.98	6.98	0	14.74
4	62.50	0	1	12.26	50	0	1	10.24
5	54.84	0	2	12.63	42.50	2.56	0	7.90

Table 4.4: Distribution of calls amongst the nurses

Probabilistic Reasoning Task	Average (ms)	CI 95%	CI 99%
Consistency	2165.43	91.18	119.83
Satisfiability	473.80	5.02	6.59
Entail Generic Stated	3030.87	107.73	141.58
Entail Generic Unstated	3995.70	82.26	108.11
Entail A-Box Stated	2508.80	38.16	50.15

Table 4.5: Performance measurements of the probabilistic reasoning tasks on an ontology with 12 probabilistic statements

4.3.2 Platform performance

4.3.2.1 Performance of the probabilistic reasoning

As mentioned in the *Priority Assessment of a call* Subsection of the *Algorithms* Section, the implementation was optimized to cope with the insufficient scalability of the probabilistic reasoning. The optimization ensures that at most 12 probabilistic statements will be extracted from the ontology on which probabilistic reasoning needs to be performed. The measurements were done using *Pronto* probabilistic *Reasoner* 0.2 on a computer with the same specifications as the previous section.

The averages and confidence intervals of all the measurements of the reasoning tasks on the ontology with 12 probabilistic statements can be seen in Table 4.5. First, the consistency and the satisfiability of the ontology were checked. Next, the performance of entailing some probabilistic statement on concept (*T-Box*) level that was explicitly stated in the ontology was checked. The performance of entailing a probabilistic statement on concept (*T-Box*) level that was not explicitly stated in the ontology was derived. *Pronto* would have to reason about the probabilistic statements to find the correct probabilistic interval. Finally, a probabilistic statement on instance (*A-Box*) level, which was explicitly stated in the ontology, was entailed. The performance is always below 4 seconds, which is acceptable.

4.3.2.2 Performance of the nurse call algorithm

Table 4.6 visualizes the performance of the different parts of the nurse call algorithm, namely assigning a staff member to a call and answering, treating (change

Call & algorithm	Average time (ms)	CI 95%	CI 99%
Normal call:			
Assign nurse	42.38	0.53	0.69
Answer call	49.79	0.50	0.65
Treat call	12.78	0.25	0.33
Finish call	65.07	0.55	0.72
Relaunch call	24.27	0.18	0.23
Sanitary call:			
Assign nurse	49.87	2.87	3.78
Answer call	54.47	3.36	4.42
Treat call	16.17	1.67	2.19
Finish call	66.71	3.97	5.21
Relaunch call	31.24	0.86	1.13
Assistance call:			
Assign nurse	57.33	2.72	3.58
Answer call	58.44	3.46	4.55
Treat call	13.18	1.57	2.07
Finish call	68.07	3.59	4.71
Relaunch call	/	/	/
Sanitary assistance call:			
Assign nurse	68.25	30.96	40.69
Answer call	54.63	12.80	16.82
Treat call	11.88	5.09	6.68
Finish call	52.63	39.11	51.40
Relaunch call	/	/	/
Urgency call:			
Assign nurse	33.83	8.69	11.42
Answer call	103.07	8.39	11.03
Treat call	7.40	3.54	4.65
Finish call	139.40	6.20	8.15
Relaunch call	/	/	/

Table 4.6: The performance results of the nurse call algorithm

status to busy) and finishing a call. Note that these results do not take into account the probabilistic reasoning to determine the priority of the call. As mentioned in the previous section, this reasoning was done in advance. When a call is launched, a suitable nurse is notified within 50.333 ms on average, which is a negligible delay.

4.4 Discussion

The first observation is that maintaining the profile information of the patients and the staff members leads to a lot of advantages.

The novel nurse call algorithm takes this information into account to intelligently assign nurses to handle calls. The place-oriented algorithm only considers which patients (actually rooms) are allocated to which nurses. In the new algorithm much more factors are taken into account. It considers the characteristics and the status of the staff members, the risk factors and preferences of the patients, the priority of the call and so on.

The nurse is able to track the location of the patient who made the call (location-awareness). Additionally the nurse also knows which patient made the call. In rooms with multiple patients, it is impossible to know accurately who made the call in the place-oriented system. In the *oNCS system*, the nurse knows specifically which patient made the call and can use this information to determine if he/she is going to answer the call or not, if medication or equipment will be needed and so on.

Even when the patient does not make a call, the nurse can access a lot of information about the patient on her PDA such as the risk factors of the patients, his or her room number and so on.

The nurse can also use the PDA to collect information about the other staff members such as their locations, if they are busy or free, which priority their task has and so on. This also makes is easier to determine if he/she is going to handle a call or not. Nurses can indicate that they are going to answer a call. In the place-oriented system unnecessary nurses are often called, which means that multiple nurses arrive at a room of a patient to handle the call. This leads to unnecessary interruptions of other tasks by these nurses. Moreover, only one nurse is called in the *oNCS system* to handle a call, while in the place-oriented system multiple nurses are often called. Additionally, it has been shown that a lot of time in hospitals is spent on trying to find someone. This will also be reduced by employing the *oNCS system*.

When a call is assigned to a nurse in the *oNCS system*, the nurse is certain that he/she is in the vicinity of the patient. Nurses that are too far away are not called to handle a call. In the place-oriented system, a nurse is sometimes called when he/she is very far away from the patient as the distance is not taken into account. The nurse cannot be sure that someone else will handle this call, which means that this nurse will have to turn back to answer the call.

When a task is interrupted, the nurse does not have to remember himself/herself that he has to return to it. The *oNCS system* does this for the nurse. This leads to fewer forgotten tasks and lesser work pressure on the staff.

A disadvantage of maintaining the profile information is the overhead that is introduced by the fact that all this information about the patients and staff members has to be inputted into the computer. Therefore this task has to be supported by a very user-friendly interface.

Secondly, the novel nurse call algorithm also leads to significant measurable improvements in the manner nurses are assigned to calls. The novel nurse call algorithm leads generally to a better workload distribution amongst the nurses as it takes into account the current task of the nurse and its priority. Additionally, only one nurse is called to handle a call, which prevents that multiple nurses arrive at a patient to handle the call. Because of this patient generally are treated quicker than in the place-oriented system. This is also caused by the fact that the distance is taken into account when searching a nurse to handle a call. Moreover, the novel nurse call algorithm takes the kind and priority of the call into account. Calls with a higher priority are generally handled faster than calls with a lower priority. This is not the case in the place-oriented system. Moreover, (sanitary) assistance calls are also generally handled faster than normal and sanitary calls. This is achieved because when a nurse receives a call while this nurse is performing a task (or even handling another call), the nurse is sure that the new call has a higher priority. This way the nurse can make a more well-funded decision on whether he/she is going to interrupt the current task or not. Moreover, the nurse is more likely to interrupt his/her task as he/she knows that this call has a higher priority and he/she is the most appropriate nurse to handle this call at this moment.

The performance of the novel nurse call algorithm is very good. A suitable nurse is notified within 50.333 ms on average, which is a negligible delay. This means that the general guidelines outlined by some countries can still be achieved. These guidelines stipulate that at least one staff member should arrive at the location of the patient within 3 minutes when an urgency call was made and within 5 minutes for normal, sanitary, service and (sanitary) assistance calls. The achieved performance does not endanger meeting these requirements.

The system scales up to at least 30 patients and 20 nurses. Thus, a lot of profile information can be retained without decreasing the performance of the system. Large-scale simulations need to be performed to profile the complete scalability of the system.

Thirdly, the portable buttons improves the mobility and the safety of the patients. Patients can walk around freely and are still able to make calls. As can be seen in the simulations it often occurs that patients need to make calls in remote areas such as smoking areas or the restaurant, where there are no nurses present. This problem is of course most prominent in departments where patients are fairly mobile e.g. the patients spend at least 10% of their time walking around.

Finally, the dynamic priority assessment of calls instead of statically defining these priorities provides a number of advantages. The priority of a call depends on the risk factors of the patients and the kind of call. This means that the priority of a call is adjusted to the specific needs and profile of the patient. This leads to a wider range of priorities of the calls that are made. This means that a patient can make calls with varying priorities depending on the current risk factors of the patient and the kind of call.

The scalability of this probabilistic assessment was presented in the *Platform Performance* Subsection of the *Results* Section. These results can be improved by calculating the probabilistic values that indicate that the patient is a low, medium or high risk patient during down-time for example at night. These are stored as facts in the ontology. This procedure does not have to repeated often as most risk factors of a patient tend not to vary that much. After doing this, the number of probabilistic statements to determine the priority of a call of a specific patient is significantly reduced to achieve an acceptable performance.

However, our study also has some limitations. A first limitation is that the probabilities in the ontology were only determined by domain experts. These probabilities indicate the probability that a patient belongs to a certain risk group based on the risk factors of this patient. A complete list of risk factors and accompanying probabilities could be constructed based on a thorough study of the risk factors of patients and the reasons for the calls that they make. However, this study is not yet conducted as the goal was to give an idea of the benefits of incorporating probabilistic priority assessment in the *oNCS system*. Probabilities were also added to the ontology to express the probability that a call of a particular kind made by a patient from a particular risk group has a particular priority. These probabilities were also determined by domain experts. In the future, the *oNCS system* could automatically learn and adapt these probabilities based on logging data from the *oNCS system*. This would make the oNCS system self-learning.

A second limitation is that the system has not been deployed in a real life environment yet. Our results are purely based on simulations. Nevertheless, these simulations were based on realistic data obtained from a department of Ghent University Hospital. However, no real observations were done in this department. The data was gathered by questioning the staff who works at the department and by examining the logging data of the current place-oriented nurse call system used in the department. This data gives us clear picture of how the patients and staff members currently move around the hospital and use the nurse call system. However, if the portable nurse call buttons would be introduced in this department, the walking behavior of the patients and nurses might change as these buttons give the patients more freedom to walk around. The usage of the nurse call buttons might also change as patients would be able to make calls from anywhere in the hospital.

Embedding a new technology into practice is not straightforward and needs to be treated with care. The adoption rate of using Information and Communication Technology (ICT) to improve the quality of care is still very low [41, 42]. One of the main reasons for this slow adoption rate is the gap in communication between the ICT and medical domain. These projects unite people with different backgrounds, such as software developers, health service researchers and nurses. Uniting all these people in a team requires effort and commitment to overcome the gap in communication. This problem can be approached by using bridge personnel who have the knowledge of multiple disciplines used in this process [43]. However, this personnel is often difficult to find.

To increase the adoption rate, the *oNCS system* can be introduced in several phases. Each phase should be supported with the needed training for the staff members and user research to adapt the system to the needs and feedback of the users.

In the first phase, the *oNCS system* software can be introduced in one department. In this phase, the new GUI is installed on the computer of the head nurse, which he/she can use to input the needed information about the staff members and patients, and the nurses are provided with PDAs to replace their portable phones or beepers. However, the mobile, portable nurse call buttons are not introduced yet to the patients. The patients keep using the nurse call buttons fixed to the walls of their room. Nevertheless, the novel nurse call algorithm is already deployed.

It is important to pick an appropriate department to test the new technology. Several criteria should be taken into account, such as openness to embrace new technology, the current usage of the nurse call system and the number of patients and nurses. It would be good to introduce the technology first in a department that would gain a lot of benefit from it. These are the departments in which there are few nurses compared to the number of patients and the patients make a reasonable amount of calls.

The most important consideration during this phase is the introduction of the GUI to the head nurse and the PDAs to the nurses. They should receive proper training to learn all the features of the GUI and the PDA. User research should also be conducted during this period, which explores the user-friendliness of the GUI and PDA. Both should be able to be customized to the preferences of the user and regular updates should be done taking the feedback of the nurses into account. It is important to emphasize to the head nurse the importance of entering all the data about the patients correctly such as their risk factors. However, a lot of the data about the patients can already be collected from the Electronic Health Record (EHR). Entering the data about the patient might seem like a tedious job for the head nurse as it introduces extra work. Therefore it is of vital importance to illustrate the benefits it introduces.

The nurses will also have to change their behavior towards receiving a call. Now they are used to often ignoring the call as multiple nurses receive it. They should be made aware that only one person receives the call at a time in the new system and that this nurse is the most appropriate person to handle the call at that time. They should only ignore it if they cannot leave their current task behind.

After the time-out time another nurse will be called. This change might not be straightforward. It is important to illustrate the advantages of the new nurse call algorithm to improve adoption. This could be done by organizing sessions between the user researchers and the nurses in which several real-life examples are shown and both nurse call algorithms are discussed. As a result the nurse call algorithm could also be updated to better suit the needs of the nurses.

When the software system is properly adopted in the first department, the second phase can start. In this phase, the portable, mobile software buttons are introduced to the patients. The patients can now freely roam through the hospital and still make calls.

This is perhaps the most invasive change. It is important to convey to the patients not to abuse the system. When they are far away from the department, they should only make calls for urgent, medical calls and not for example for a glass of water. Otherwise nurses might have to walk long distances to answer simple calls, which might be rather frustrating.

Nurses can now also be called for patients who are not in their department e.g. because a patient becomes unwell inside a staircase far away from his/her own department. The implications of this should be thoroughly studied e.g. rules for responsibilities for patients.

In the third phase, the *oNCS system* can be gradually introduced into other departments of the hospital. The adoption rate in these other departments should be quicker, as the system has been thoroughly tested in the first department. Moreover, this department can be used as an illustration of the advantages of the system.

4.5 Conclusion

This article showed that the current nurse call algorithms could be significantly improved by storing profile information about the staff members and patients in an ontology. Moreover, it introduces a software system that could easily be used to introduce portable nurse call buttons, which improve the mobility of patients, location-awareness and safety.

The person-oriented nature of the platform was clearly illustrated by using the context information about the risk factors of a patient to dynamically determine the priority of the call this patient is making. By using probabilistic reasoning algorithms, the probability that a specific call made by a specific patient has a specific priority can be determined. These probabilities are derived from the different risk factors of this patient as these risk factors will influence the probability that a patient makes urgent calls. All these probabilistic values are combined in an intelligent manner to determine the most suitable priority for this call.

The benefits of this novel *oNCS system* are illustrated with realistic simulations about data collected from the Ghent University Hospital. The *oNCS system* significantly improves the assignment of nurses to calls. Calls generally have a nurse present faster, the workload-distribution amongst the nurses improves and the priorities and kinds of the calls are taken into account. The execution time of the nurse call algorithm is negligible. However, before the system can be widely deployed, it is important that first a thorough study is done to characterize the correlation between the risk factors of patients and the reasons for their calls.

Future work will mainly focus on improving the scalability of the probabilistic assessment algorithm to determine the priority of a call. Simultaneously, hardware and algorithms for the effective and accurate determination of the location of staff members and patients will be further studied. Finally, the performance and benefits of the system will be thoroughly studied by performing realistic tests on the large-scale sensor network available within the IBCN research group.

Acknowledgment

We would like to acknowledge that this research was supported by the excellent information given by the head nurse Annemie Danhieux about the inner workings of her department at Ghent University Hospital. Special thanks also go out to my colleague Matthias Strobbe for supporting the implementation in the CASP platform. Finally, we would like to acknowledge the people behind Pronto and especially Pavel Klinov for always being open to questions and discussions.

Femke Ongenae is funded by a PhD grant of the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT).

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

FO and DM carried out the study, participated in the development of the concepts described in this paper and drafted the manuscript. JD and PV participated in the case study. FDT, TD, TD and DVG supervised the study, participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

References

[1] S. V. Hoecke, J. Decruyenaere, C. Danneels, K. Taveirne, K. Colpaert, E. Hoste, B. Dhoedt, and F. D. Turck. *Service-oriented subscription management of medical decision data in the intensive care unit*. Methods of Information in Medicine, 47(4):364–380, 2008.

[2] K. Colpaert, S. V. Belleghem, D. Benoit, K. Steurbaut, F. D. Turck, and J. Decruyenaere. *Has information technology finally been adopted in intensive care units?* In 22nd Annual Congress of the European Society of Intensive Care Medicine, page 235, Vienna, Austria, October 11-14 2009.

[3] M. Tentori, D. Segura, and J. Favela. *Chapter VIII: Monitoring hospital patients using ambient displays*. In P. Olla and J. Tan, editors, Mobile Health Solutions for Biomedical Applications, volume 1, pages 143–158. Medical Information Science Reference;, Hershey, New York, USA, 1st edition, 2009.

[4] G. W. Wachter. *Hospitals Unplugged: The Wireless Revolution Reaches Healthcare*. Telemedicine Information Exchange, 2001.

[5] E. T. Miller, C. Deets, and R. Miller. *Nurse call and the work environment: lessons learned*. J Nurs Care Qual, 15(3):7–15, 2000.

[6] L. Linden and K. English. *Adjusting the cost-quality equation: Utilizing work sampling and time study data to redesign clinical practice*. J Nurs Care Qual, 8(3):34–42, 1994.

[7] E. T. Miller, C. Deets, and R. Miller. *Nursing call systems: Impact on nursing performance*. J Nurs Care Qual, 11(3):36–43, 1997.

[8] C. M. Meade, A. L. Bursell, and L. Ketelsen. *Effect of nursing rounds on patients' call light use, satisfaction and safety*. AM J NURS, 106(9):58–70, 2006.

[9] T. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199–220, 1993. Available from: http://ontology.csse.uwa.edu.au/reference/browse_paper.php?pid=233281545, doi:10.1.1.101.7493.

[10] D. Preuveneers, J. V. d. Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. D. Bosschere. *Towards an extensible context ontology for Ambient Intelligence*. In P. Markopoulos, B. Eggen, E. Aarts, and J. L. Crowley, editors, 2nd European Symposium on Ambient Intelligence (EUSAI 2004), volume 3295/2004 of *Lecture Notes in Computer Science*, pages 148–159, Eindhoven, The Netherlands, Nov 8–11 2004. Springer Berlin / Heidelberg. doi:10.1007/978-3-540-30473-9_15.

[11] T. Gu, H. K. Pung, and D. Q. Zhang. *Toward an OSGI-Based Infrastructure for Context-Aware Applications*. Journal of Pervasive Computing, IEEE, 3(4):66–74, October 2004.

[12] H. Chen, T. Finin, and A. Joshi. *An ontology for context-aware pervasive computing environments*. The Knowledge Engineering Review, 18(3):197–207, 2004.

[13] A. Anjum, P. Bloodsworth, A. Branson, T. Hauer, R. McClatchey, K. Munir, D. Rogulin, and J. Shamdasani. *The requirements for ontologies in medical data integration: A case study*. In 11th International Database Engineering & Applications Symposium, pages 308–314, September 6-8 2007.

[14] B. Smith and M. Brochhausen. *Putting biomedical ontologies to work*. Methods of Information in Medicine, 49(2):135–140, 2010.

[15] M. Becker, C. Heine, R. Herrler, and K. H. Krempels. *Chapter VII: An Ontology for Hospital Scenarios*. In A. Moreno and J. L. Nealon, editors, Applications of Software Agent Technology in the Health Care Domain, pages 87–103. Birkhäuser Basel, 1st edition, 2003. doi:10.1007/978-3-0348-7976-7.

[16] P. Kataria, A. Macfie, R. Juric, and K. Madani. *Ontology For Supporting Context Aware Applications For The Intelligent Hospital Ward*. Journal of Integrated Design & Process Science, 12(3):35–44, 2008.

[17] W. Yao, C. Chu, A. Kumar, and Z. Li. *Using Ontology to Support Context Awareness in Healthcare*. In 19th Workshop on Information Technologies and Systems, December 14-15 2009.

[18] M. Strobbe, J. Hollez, G. D. Jans, O. V. Laere, J. Nelis, F. D. Turck, B. Dhoedt, P. Demeester, N. Janssens, and T. Pollet. *Design of CASP: an open enabling platform for context aware office and city services*. In T. Pfeifer, J. Strassner, and S. Dobson, editors, Proceedings of the 4th International Workshop on Managing Ubiquitous Communications and Services (MUCS 2007), pages 123–142, Munich, Germany, May 25 2007. ISBN:3-930736-07-1. Available from: http://en.scientificcommons.org/23102335.

[19] *Ghent University hospital*. http://www.healthcarebelgium.com/index.php?id=uzgent, 2013.

[20] X. An, J. Wang, R. V. Prasad, and I. G. M. M. Niemegeers. *OPT: online person tracking system for context-awareness in wireless personal network*. In International Symposium on Mobile Ad Hoc Networking & Computing, pages 47–54, May 22-25 2006.

[21] T. A. Alhmiedat and S. Yang. *A ZigBee-based mobile tracking system through wireless sensor networks*. IJAMECHS, 1(1):63–70, 2008.

[22] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: concepts, architectures and applications*. Springer Verlag, Berlin, Germany, 2003.

[23] *Televic NV, specialized in nurse call systems, audio and multimedia communication*. http://www.televic.com, 2013.

[24] F. Ongenae, M. Strobbe, J. Hollez, G. D. Jans, F. D. Turck, T. Dhaene, P. Demeester, and P. Verhoeve. *Ontology based and context-aware hospital nurse call optimization*. In F. Xhafa and L. Barolli, editors, CISIS 2008: THE SECOND INTERNATIONAL CONFERENCE ON COMPLEX, INTELLIGENT AND SOFTWARE INTENSIVE SYSTEMS, PROCEEDINGS, pages 985–990. IEEE COMPUTER SOC, 2008.

[25] F. Ongenae, M. Strobbe, J. Hollez, G. D. Jans, F. D. Turck, T. Dhaene, P. Demeester, and P. Verhoeve. *Design of a semantic person-oriented nurse call management system*. INTERNATIONAL JOURNAL OF WEB AND GRID SERVICES, 4(3):267–283, 2008.

[26] P. Klinov. *Pronto: A Non-monotonic Probabilistic Description Logic Reasoner*. In Proceedings of the 5th European Semantic Web Conference (ESWC), pages 822–826, Tenerife, Spain, June 1-5 2008. Available from: http://pellet.owldl.com/pronto.

[27] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003. ISBN:0521150116.

[28] T. Lukasiewicz. *Probabilistic Description Logics for the Semantic Web*. Technical report, Technical University of Wien, Institute for Information Systems, 2007.

[29] P. Klinov and B. Parsia. *Optimization and Evaluation of Reasoning in Probabilistic Description Logic: Towards a Systematic Approach*. In 7th International Semantic Web Conference, pages 213–228, October 26-30 2008.

[30] D. L. McGuinness and F. v. Harmelen. *OWL Web Ontology Language Overview*. Technical Report REC-owl-features-20040210, World Wide Web Consortium, http://www.w3.org/TR/owl-features/, Online, February 10 2004.

[31] *W3C: World Wide Web Consortium*. http://www.w3.org/, 2013.

[32] E. Prud'hommeaux and A. Seaborne. *SPARQL Query Language for RDF*. Technical report, World Wide Web Consortium, Recommendation REC-rdf-sparql-query-20080115, http://www.w3.org/TR/rdf-sparql-query/, Online, January 15 2008.

[33] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. *Jena: implementing the semantic web recommendations*. In Proceedings of the 13th international conference on World Wide Web, Alternate track papers & posters (WWW Alt. 2004), pages 74–83, New York, NY, USA, May 17–22 2004. ACM. doi:10.1145/1013367.1013381. Available from: http://portal.acm.org/citation.cfm?doid=1013367.1013381, doi:http://doi.acm.org/10.1145/1013367.1013381.

[34] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, The Semantic Web – 3rd International Semantic Web Conference (ISWC 2004), volume 3298/2004 of *Lecture Notes in Computer Science*, pages 229–243. Springer Berlin / Heidelberg, Hiroshima, Japan, November 7–11 2004. doi:10.1007/978-3-540-30475-3_17. Available from: http://dx.doi.org/10.1007/978-3-540-30475-3_17.

[35] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL reasoner*. Journal of Web Semantics, 5(2):51–53, 2007. doi:10.1016/j.websem.2007.03.004. doi:10.1016/j.websem.2007.03.004.

[36] P. F. Patel-Schneider, I. Horrocks, and B. C. Grau. *OWL 1.1 Web Ontology Language Overview*. Technical report, World Wide Web Consortium, http://www.w3.org/Submission/2006/SUBM-owl11-overview-20061219/, Online, December 19 2006.

[37] S. Haiges. *A Step By Step Introduction to OSGi Programming Based on the Open Source Knopflerfish OSGi Framework*. Technical report, October 2004.

[38] L. B. Sokolinsky. *Survey of architectures of parallel database systems*. PROGRAM COMPUT SOFT+, 30(6):337–346, 2004.

[39] E. Rahm. *Dynamic load balancing in parallel database systems*. In 2nd International Parallel Processing Conference, pages 37–52, August 26-29 1996.

[40] D. Myny, D. V. Goubergen, V. Limere, M. Gobert, S. Verhaeghe, and T. Defloor. *Determination of standard times of nursing activities based on a Nursing Minimum Dataset*. J ADV NURS, 66(1):92–102, 2010.

[41] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach. *Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success*. British Medical Journal, 330(7494):765, 2005.

[42] A. Garg, N. Adhikari, H. McDonald, P. Rosas-Arellano, P. Devereaux, J. Beyene, J. Sam, and B. Haynes. *Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review*. Journal of the American Medical Association, 193(10):1223–1238, 2005.

[43] R. Shiffman, G. Michel, A. Essaihi, and E. Thornquist. *Bridging the Guideline Implementation Gap: A Systematic, Document-Centered Approach to Guideline Implementation*. Journal of the American Medical Informatics Association, 11(5):418–426, 2004.

5

A Probabilistic Ontology-based Platform for Self-learning Context-aware Healthcare Applications

F. Ongenae, M. Claeys, T. Dupont, W. Kerckhove, P. Verhoeve, Tom Dhaene and F. De Turck

Accepted for Expert Systems with Applications, July 2013.

"The context-aware system should not be a strict and controlling Big Brother, but a wise, understanding and caring Big Mother."

– Piet Verhoeve (2007)

When new technology is introduced, the behavior of the users changes to adapt to it. Moreover, different environments in which the application is deployed might have slightly different requirements pertaining to how the context information is taken into account. It is difficult to foresee these changes at development time. Therefore, a self-learning framework was developed, which allows context-aware healthcare applications to adapt their behavior at run-time and achieve truly personalized healthcare services. A thorough evaluation of the applicability, correctness and performance of the self-learning framework in the healthcare domain *was performed by an illustrative scenario concerning the discovery of reasons for patients' call light use. For this evaluation the continuous care ontology was used, which was presented in Chapter 2. In this chapter the research is discussed that was performed to realize Research Contribution 4 highlighted in Section 1.3 of Chapter 1.*

⋆ ⋆ ⋆

Abstract Context-aware platforms consist of dynamic algorithms that take the context information into account to adapt the behavior of the applications. The relevant context information is modeled in a context model. Recently, a trend has emerged towards capturing the context in an ontology, which formally models the concepts within a certain domain, their relations and properties.

Although much research has been done on the subject, the adoption of context-aware services in healthcare is lagging behind what could be expected. The main complaint made by users is that they had to significantly alter workflow patterns to accommodate the system. When new technology is introduced, the behavior of the users changes to adapt to it. Moreover, small differences in user requirements often occur between different environments where the application is deployed. However, it is difficult to foresee these changes in workflow patterns and requirements at development time. Consequently, the context-aware applications are not tuned towards the needs of the users and they are required to change their behavior to accommodate the technology instead of the other way around.

To tackle this issue, a self-learning, probabilistic, ontology-based framework is proposed, which allows context-aware applications to adapt their behavior at run-time. It exploits the context information gathered in the ontology to mine for trends and patterns in the behavior of the users. These trends are then prioritized and filtered by associating probabilities, which express their reliability. This new knowledge and their associated probabilities are then integrated into the context model and dynamic algorithms. Finally, the probabilities are in- or decreased, according to context and behavioral information gathered about the usage of the learned information.

A use case is presented to illustrate the applicability of the framework, namely mining the reasons for patients' nurse call light use to automatically launch calls. Detecting Systemic Inflammatory Response Syndrome (SIRS) as a reason for nurse calls is used as a realistic scenario to evaluate the correctness and performance of the proposed framework. It is shown that correct results are achieved when the dataset contains at least 1000 instances and the amount of noise is lower than 5%. The execution time and memory usage are also negligible for a realistic dataset, i.e., below 100 milliseconds (ms) and 10 megabyte (MB).

5.1 Introduction

Computerized tools, health monitoring devices and sensors are being actively adopted in modern healthcare settings, especially to support administrative tasks, data management and patient monitoring [1, 2]. Today, caregivers are directly faced with these technologies, which increases the complexity of their daily activities [3]. The caregiver has to use several devices to manually consult, insert and combine data, even when carrying out a single task. This is very time-consuming. Due to this inadequate integration of the technology, as well as the large amount of data being generated by the devices and the heavy workload of staff members, it is not rare for important events to be missed, e.g., early indications of worsening condition of a patient. To resolve this issue, context-aware techniques are often proposed to automatically exploit the medical information available to improve continuous care and personalize healthcare [4].

Although much research has been done on the subject, the adoption of context-aware services is lagging behind what could be expected. Most of the projects are prototypes and real applications are still difficult to find. Whereas the healthcare industry is quick to exploit the latest medical technology, they are reluctant adopters of modern health information systems [5]. Half of all computer-based information systems fail due to user resistance and staff interference [6]. The main complaint made against mobile, context-aware systems is that users had to significantly alter workflow patterns to accommodate the system [7]. This is due to inadequate techniques for personalization of the services, a lack of focus on the soft aspects of interaction, e.g., automated and personalized alerts, and the lack of tackling problems such as the need of the users for control [8].

The context-aware platforms use dynamic algorithms, which take the context information into account, to adapt the behavior of the applications according to the context and offer personalized services to the users. However, these algorithms are defined at development time. When new technology is introduced, the behavior of the users changes to adapt to it. Moreover, different environments in which the application is deployed, e.g., different nursing units or hospital departments, might have slightly different requirements pertaining to how the context information is taken into account. It is difficult to foresee these changes in behavior and small nuances in workflows at development time. This means that the context model might be incomplete or the algorithms of the applications built on it may no longer apply. As the applications do not adapt to the requirements and workflow patterns of the users, they feel less in control of the technology and have to adapt their behavior to accommodate the technology instead of the other way around.

To tackle this issue, this paper proposes a self-learning framework, which allows the context-aware applications to adapt their behavior at run-time to accommodate the changing requirements of the users. The proposed framework consist of the following techniques. First, an ontology-based context model with accompanying rule-based context-aware algorithms is used to capture the behavior of the user and the context in which it is exhibited. This captured information is then filtered, cleaned and structured so that it can be used as input for data mining techniques. The results of these data mining techniques are then prioritized and filtered by associating probabilities with the obtained results expressing how reliable or accurate they are. These results and their associated probabilities are then integrated into the context model and dynamic algorithms. These probabilities clarify to the stakeholders that this new knowledge has not been confirmed by rigorous evaluation. Finally, the probabilities are adapted, i.e., in- or decreased, according to context and behavioral information gathered about the usage of the learned information.

The remainder of this article is organized as follows. In Section 5.2 the relevant related work is discussed and our contribution is highlighted. Section 5.3 presents the architecture of the proposed probabilistic ontology-based framework for self-learning context-aware healthcare applications. Section 5.4 discusses the generic implementation of the framework, i.e., the classes that can be extended to implement the specific use cases. The implementation of a specific use case, namely mining the reasons for patients' call light use to automatically launch calls, is presented in Section 5.5. Finally, the main conclusions of this research are highlighted and the future work is discussed in Section 5.6.

5.2 Related work

5.2.1 Context-aware systems

Dey and Abowd [9] refer to context as "any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves". A system may be labeled as "context-aware" if it can acquire, interpret and use context information to adapt its behavior to the current context in use [10]. A number of generic context platforms have been developed to relieve application developers from the aggregation and abstraction of context information and the derivation of high-level contexts [11–14]. Unorganized, unprocessed raw data can be voluminous, but has no meaning on itself as it has no relationships or context. Information is data that has been given meaning by defining relational connections. The proposed platforms employ several techniques to model this context information, i.e., key-value, markup scheme, graphical, object-oriented, logic-based and ontology-based models [15]. A notable trend is emerging towards ontology-based context-aware platforms [16–19].

To write the dynamic algorithms, which take the context information captured in the ontology into account to achieve personalized and context-aware applications, two approaches are commonly used, namely rules or machine learning techniques [20]. Rules are manually constructed at development time and thus require developers to foresee all possible situations that can occur at runtime and define the appropriate corresponding actions. Rules are difficult to modify, maintain and scale [21]. Machine learning techniques, e.g., Bayesian networks and neural networks, are also trained at development time. Bayesian networks suffer from similar maintenance and scalability problems as the rule-based approach and acquiring accurate probabilities is a tedious job [22]. Neural Networks require a lot of processing power and have consequently only been sparsely applied in context-aware applications. Their black-box nature also makes it difficult to gain insight into relations between context and actions, increasing the fear of technology and loss of control from the users. Consequently, with each of these approaches, the context-aware system is only able to cope with a fixed set of context changes that were taken into accounted during the design of the system.

As mentioned previously, run-time adaptation of the dynamic algorithms is needed to adapt to changing behavior of the stakeholders and to truly offer personalized services tuned to the work practices of the specific environment where the application is deployed. A couple of context-aware systems exist that try to tackle this problem by mining historical information [20, 23–25]. However, most of the research focusses on the development of data mining techniques, which can be used to learn the patterns and requirements, or use a black-box approach. Litte research has been done on the development of a complete framework for self-learning, context-aware applications and on how the learned knowledge should be integrated in an ontology-based platform.

5.2.2 Context-aware systems in healthcare

The use of context and context-awareness in healthcare is an active research area [26, 27]. First, there is a large amount of available information, specific healthcare situations and related tasks, which create a potential for cognitive overload amongst the caregivers. Second, the patients, healthcare professionals and some equipment are fairly mobile, which requires accurate localization and adaptation of the healthcare services to the environment. Third, the financial and human resources are limited. This implies a need to cut cost while improving the quality of service to an increased number of people. Context-aware and pervasive prototypes have been developed for a number of hospital [28–32] and homecare & residential care [33–40] use cases. Examples of context-aware healthcare systems based on ontologies can also be found in literature [41–44].

5.2.3 eHealth ontologies

An ontology [45] is a semantic model that formally describes the concepts in a certain domain, their relationships and attributes. In this way, an ontology encourages re-use and integration. By managing the data about the current context in an ontology, intelligent algorithms that take advantage of this information to optimize and personalize the context-aware applications, can more easily be defined. The Web Ontology Language (OWL) [46] is the leading language for encoding these ontologies. Because of the foundation of OWL in Description Logics (DLs) [47], which are a family of logics that are decidable fragments of first-order logic, the models and description of data in these models can be formally proved. It can also be used to detect inconsistencies in the model as well as infer new information out of the correlation of this data. This proofing and classification process is referred to as Reasoning. Reasoners are implemented as generic software-modules, independent of the domain-specific problem. Ontologies thus effectively separate the domain knowledge, which can be re-used across different applications, from the application logic, which can be written as rules on top of the ontology.

The definition and use of ontologies in the medical domain is an active research field, as it has been recognized that ontology-based systems can be used to improve the management of complex health systems [48]. Most of the developed ontologies focus on biomedical research and are mainly employed to clearly define medical terminology [49], e.g., Galen Common Reference Model [50], the Foundational Model of Anatomy Ontology (FMA) [51] or the Gene Ontology [52]. Little work has been done on developing high-level ontologies, which can be used to model context information and knowledge utilized across the various continuous care settings [53]. However, ontologies have been developed for specific subdomains of continuous care, e.g., ontologies for structuring organization knowledge in homecare assistance [48], representing the context of the activity in which the user is engaged [54] and modeling chronic disease management in homecare settings [43].

5.2.4 Our contribution

In this paper, we propose a self-learning and probabilistic framework to adapt the behavior of ontology-based, context-aware applications to the changing requirements of the users and their workflow patterns. To our knowledge, little previous research has been done on how discovered trends and patterns can be integrated into ontology-based platforms without making the existing model inconsistent. To tackle this issue, we use a probabilistic approach, which conveys the reliability of the learned knowledge to the users and ensures the compatibility with existing knowledge in the context model. Moreover, the existing research on self-learning, context-aware applications concentrates on exploring data mining techniques, which can be used to discover the trends and patterns. Our research focuses on the development of a complete framework to enable self-learning, context-aware healthcare applications.

5.3 Architecture of the self-learning, context-aware framework

The general architecture of the proposed self-learning, context-aware framework is visualized in Figure 5.1. The following subsections discuss the different components and modules of this framework in more detail.

5.3.1 Context-aware platform

The general architecture of a context-aware, ontology-based platform can be split up into five layers. The *Device Layer* includes all the devices and the software on those devices that deliver context information. The modern healthcare settings contains a plethora of computerized medical equipment to convey the condition of a patient, e.g., monitoring equipment, electronic patient records and laboratory results stored in a database, and support the caregivers in their daily activities, e.g., nurse call systems and task management and planning tools.

The *Context Provider Layer* takes care of the acquisition of specific context information, e.g., location or presence information, and translates it to ontology instances. These ontology instances are then added to the *Knowledge Base* in the *Semantic Reasoning Layer*. This *Knowledge Base* aggregates all the relevant context information into a formal context model, i.e., an ontology. Existing *Medical and Context-Aware Ontologies* are integrated into the platform and extended with *Domain Ontologies* which model the information specific to a particular healthcare setting, e.g., the specific roles and competences of the caregivers and how they map on each other, the available monitoring equipment and their threshold values and specific tasks that need to be performed. These *Domain Ontologies* can also contain probabilistic information, e.g., a call made by patient with a heart disease has 25% chance of being urgent.

Reasoning components are then used to derive new, high-level knowledge from the information aggregated in the *Knowledge Base*. Due to the foundation of ontologies in *Description Logics* (*DL*), the models can be formally proofed by using a *DL Reasoner*. This *DL Reasoner* is used to detect inconsistencies in the model as well as infer new information from the correlation of the data. For example, a concept *Fever* is created in the ontology, which automatically detects patients with a temperature above 38 °C. More complex logic is expressed by defining *Rules* on top of this ontology and performing *Rule-based Reasoning*.

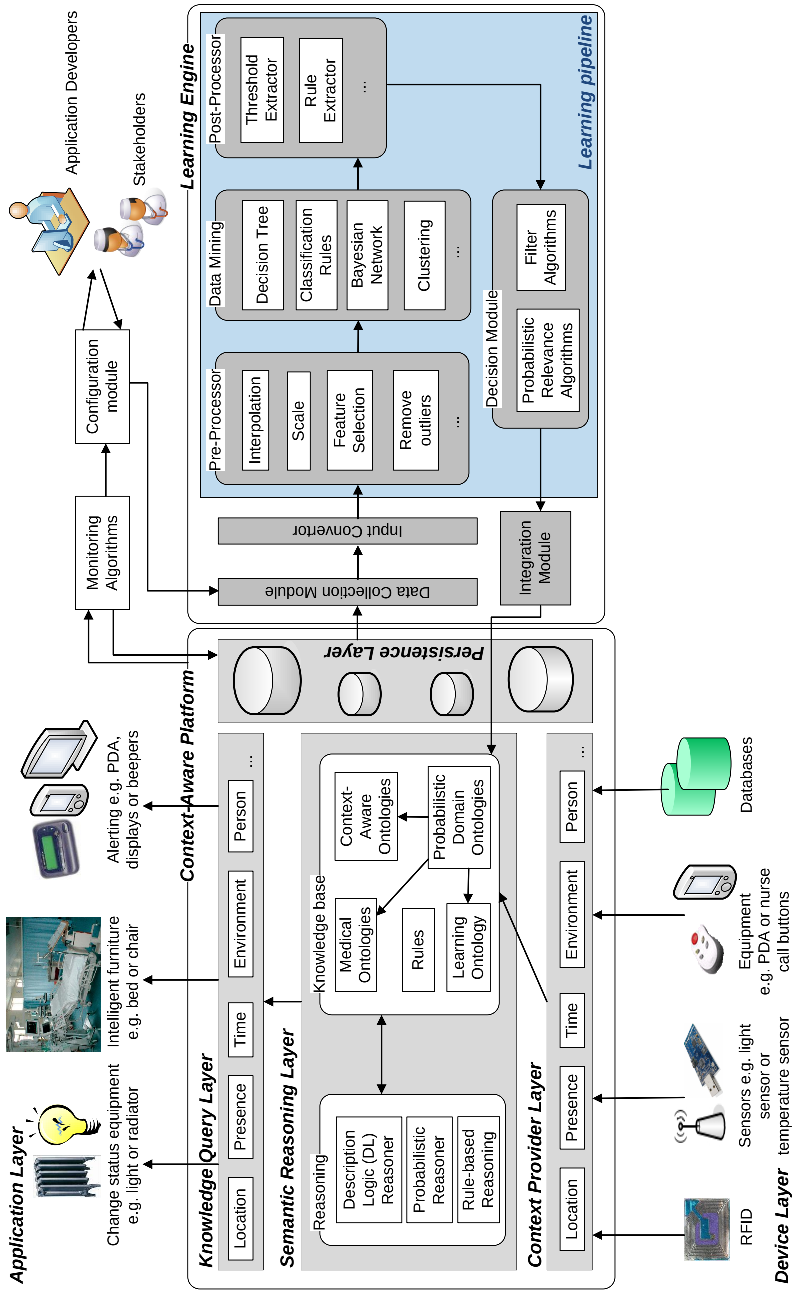


Figure 5.1: General architecture of the self-learning, context-aware framework

The *Knowledge Query Layer* facilitates the retrieval of context information such that it can be used by the different applications and services. The *Application Layer* includes all the devices and the software on those devices that use the (derived) context information to adapt their behavior.

Finally, the *Persistence Layer* ensures the persistence of context information. Static contextual information about users, devices and the environment can be easily obtained from these databases. More importantly, the *Persistence Layer* can also be used to store more dynamic information, such as previous locations of caregivers and patients or actions taken by the users.

The *Semantic Reasoning* and *Persistence Layers* are the most important layers to facilitate a self-learning *Context-Aware Platform*. As the *Knowledge Base* integrates all the context information, it gives insight into the behavior and changing requirements of the users. All the collected context information and the knowledge derived from it is then persisted in the databases from the *Persistence Layer*. This lets the *Learning Engine* exploit this history of context information to derive trends and patterns and adapt the information in the ontology and accompanying rules accordingly.

5.3.2 Monitoring algorithms and configuration module

Monitoring Algorithms determine missing or inaccurate knowledge in the ontology. An example: situations are logged where a suggestion is given by the system to the staff to do an action, but under certain circumstances the caregivers consistently execute a different action. The *Monitoring Algorithms* constantly monitor the ontology for interesting situations. They gather these situations and store them collectively in the *Persistence Layer*. The results of the *Monitoring Algorithms* can intermediately be shown to *Stakeholders*, i.e., domain experts such as nurses, doctors and professionals working for the healthcare industry, and *Application Developers*. When enough data has been collected, the *Learning Engine* can be initiated. The amount of data that should be gathered depends on the specific use case and the used data mining technique. The input parameters are specified in the *Configuration Module* and the *Data Collection Module* automatically extracts the appropriate data from the *Persistence Layer*. The *Configuration Module* is also responsible for configuring the pipeline. A default pipeline can be used or a specific configuration can be indicated by the *Stakeholders* or *Application Developer*s.

Note, that the *Configuration Module* can be configured both by the *Monitoring Algorithms* themselves and by the *Stakeholders & Application Developers*. It can thus be regulated how much autonomy the *Learning Engine* has. Moreover, the possibility of human intervention avoids unnecessary learning steps in case the new knowledge, which should be added to the ontology based on the observation from the *Monitoring Algorithms*, is straightforward. Finally, the results of the

Monitoring Algorithms give the *Stakeholders & Application Developers* insight into the behavior and requirements of the users.

5.3.3 Learning engine

The Pipes-and-Filters architectural design pattern [55] was used to design the *Learning Engine*. This data-driven pattern divides a larger processing task into a sequence of smaller, independent processing steps, called filters, that are connected by channels, called pipes. Each filter provides a simple interface, namely it receives messages on the incoming pipe, processes them and provides the results to the outgoing pipe. A filter is thus unaware of its position in the pipeline and which filter precedes and follows it. Because all the filters use similar interfaces they can be combined into different pipelines. Filters can thus easily be added, omitted or rearranged. As a result, the architecture becomes very modular, extensible, re-usable and flexible.

5.3.3.1 Data collection and input conversion

To be able to use a flexible Pipes-and-Filters architecture, the data exchanged between the filters needs to be expressed in the same format. A format was developed, which allows expressing both the information which is used as input and the knowledge that is obtained as output, e.g., rules. The format is largely based on the Attribute-Relation File Format (ARFF), which is the text file format used by WEKA [56].

The Data Collection Module is responsible for gathering the necessary input information for the *Learning Engine* from the *Persistence Layer*. The *Input Convertor* converts this data to the data format used by the *Learning Pipeline*. The *Data Collection Module* and *Input Convertor* cannot be considered as actual filters for two reasons. First, for any use case scenario they will always appear as the first two steps of the pipeline. Second, the input and output format of these modules is dependent on the source from which the information is collected, e.g., a triple store.

5.3.3.2 Learning pipeline

The *Pre-Processor* contains several modules to clean up the data. For example, the *Remove Outliers* component removes unrealistic entries from the input data, e.g., impossible sensor values. The *Scale* component centers the input values at zero. This is often beneficial for the learning algorithms of various machine learning techniques. *Feature Selection* can be used to reduce the size of the input data set and thus speed up the data mining. Other examples of pre-processing techniques can easily be integrated into the pipeline as new Filters.

The cleaned data is then passed to the *Data Mining* component that provides several techniques to discover trends, e.g., classification rules, decision trees, Bayesian networks or clustering. The results of the *Data Mining* are then processed by the *Post-Processor* to derive the actual information which can be added to the ontology, e.g., rules or thresholds can be derived from a decision tree by the *Rule* or *Threshold Extractor*.

The conclusions of the *Post-Processor* are studied further by the *Decision Module*. To ensure that the *Knowledge Base* does not become inconsistent when the new knowledge is added, i.e., because it contradicts with already defined knowledge, probabilistic relations are defined between the new and existing knowledge. Moreover, this probability also makes clear to the *Stakeholders* that the new knowledge has not been confirmed by rigorous evaluation yet. The *Probabilistic Relevance Algorithms* are used to determine the initial probability that should be associated with this new knowledge. For example, it can be calculated how many times a derived rule occured in the data set on which the data mining technique was trained. However, wrong trends can easily be detected because of skewed or too small data sets. It is also important to only include trends that reflect good and general work practices. Wrong information could clutter the *Knowledge Base* and make the context-aware platform less useable. The *Filter Algorithms* are responsible for detecting and removing these anomolies, e.g., by removing knowledge that received a too low probability by the *Probabilistic Relevance Algorithms*.

The *Learning Pipeline* cannot only be used to learn new information, but also to reassess knowledge that has been previously added to the *Knowledge Base*. In this case, the *Probabilistic Relevance Algorithms* are responsible for in- or decreasing the probability depending on the new information that becomes available about the usage of this knowledge.

5.3.3.3 Integration module and adapting the probabilities

Finally, the *Integration Module* is responsible for defining the probabilistic relations that connect the new knowledge to the existing knowledge in the *Knowledge Base*. For the same reasons as were already explained in Section 5.3.3.1 for the *Data Integration* and *Input Convertor Modules*, this module cannot really be considered a filter.

For new knowledge, the probability calculated by the *Probabilistic Relevance Algorithms* is used. When the *Stakeholders* are confronted with a probabilistic decision in their daily work practices, they might be interested in the origin of the information, i.e., how the information was learned, before deciding to follow the recommendation of the context-aware platform or not. Therefore, the *Learning Ontology* was created, which allows associating the learned knowledge with its origin. The most important concepts of this ontology are visualized in Figure 5.2. This ontology also allows *Application Developers* to easily identify learned knowl-

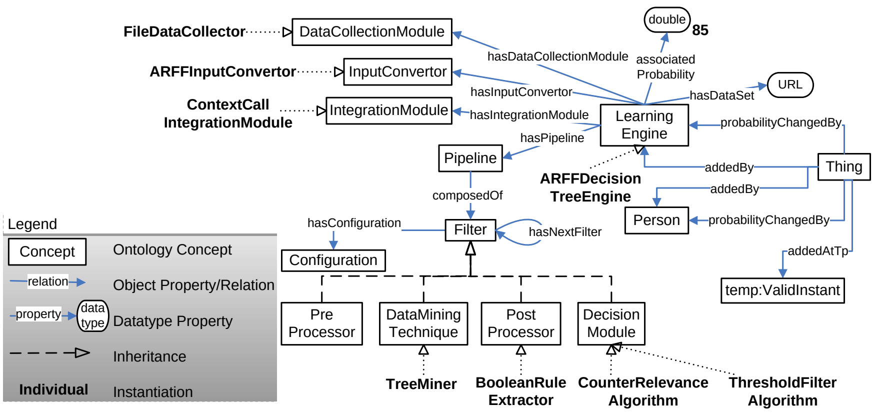


Figure 5.2: The Learning Ontology

edge. This enables them to treat this knowledge differently if needed, e.g., ignore it in reliability critical applications or highlight it for the users.

For reassessed knowledge, two thresholds are checked. If the probability calculated by the *Probabilistic Relevance Algorithms* falls below the lowest threshold, the knowledge is removed from the *Knowledge Base* as it is clearly not being used or confirmed by the stakeholders. If the probability exceeds the highest threshold, the knowledge is added to the ontology as generally accepted knowledge, i.e., without an associated probability. Finally, if the probability lies between the two thresholds, the probabilitiy of the reassessed knowledge is updated to this probability to reflect its changed reliability. As such, a self-learning, context-aware platform is obtained in which knowledge can be added and removed on the fly based on historical information.

5.4 Implementation details

The implementation details of the *Context-Aware Platform* are described in Strobbe, et al. [57, 58]. The platform uses OWL [46] as ontology language, Pellet [59] as *DL Reasoner*, Jena Rules [60] and SWRL [61] to express the *Rules* and SPARQL [62] to query the context information. The platform was extended with the *Probabilistic Reasoner* Pronto [63] to enable probabilistic reasoning on the ontologies. Jena is used to manage and persist the ontologies.

The *Learning Engine*, *Monitoring Algorithms* and *Configuration Module* were implemented in Java. The class diagram of the *Learning Engine* is visualized in Figure 5.3. These are the (abstract) classes, which can be used for any scenario. To implement a specific use case, subclasses can be created that implement the specific requirements of the scenario, e.g., a specific pipeline configuration or a

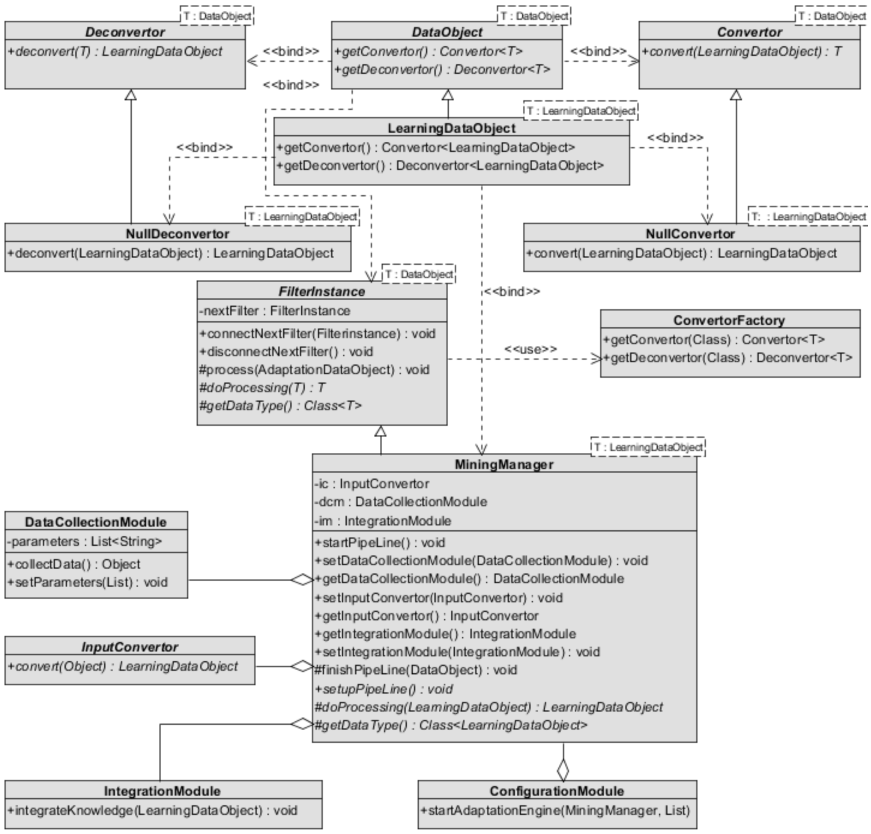


Figure 5.3: The class diagram of the Learning Engine and Configuration Module

specific input convertor. An example of how a specific use case can be implemented is thoroughly explained in Section 5.5. How these classes can be used to construct and use a specific *Learning Pipeline* with associated *Data Collection Module*, *Input Convertor* and *Integration Module* is visualized with a sequence diagram in Figure 5.4.

As can be seen, the different filters in the *Learning Pipeline* are respresented by `FilterInstance` objects. Specific filters, e.g., pre- and post-processors, filter algorithms and data mining techniques, are created as subclasses of this `FilterInstance` class by implementing the `doProcessing` method. This method specifies how the data is processed by the specific filter, e.g., a `ScalingFilter` that scales the data or a `ClusterFilter` that clusters it.

As mentioned previously, the data exchanged between the filters in the pipeline uses the same data format, which is represented by the `LearningDataObject` Java-Object. This object contains the information about the different attributes,

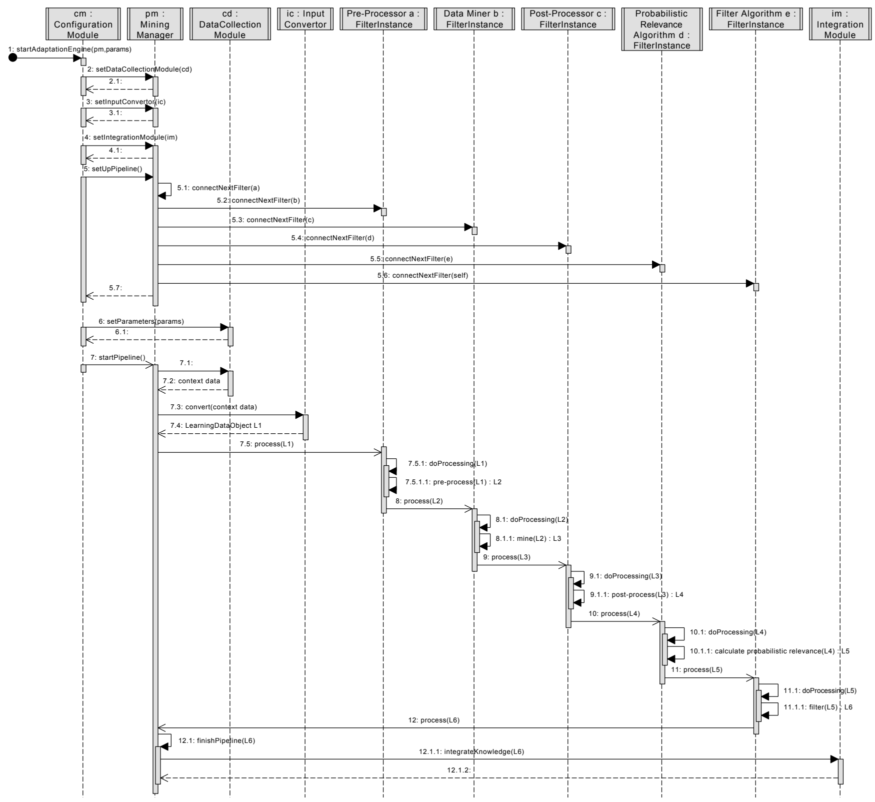


Figure 5.4: Sequence diagram illustrating the construction and usage of a Learning Pipeline with associated Data Collection Module, Input Convertor and Integration Module

i.e., ontology concepts, which will be mined, and their data instances. However, to enable logging of the data at any point during the pipeline, this object can easily be serialized to XML.

As can be seen, (de)convertors can be used to translate the specific data format to other formats. This is not only necessary to convert the context data gathered by the *Context-Aware Platform* to the data format used by the pipeline, but also to allow the usage of external libraries, e.g., WEKA for data mining. The (de)convertors allow to transform the `LearningDataObject` to the format used by the external libraries, e.g., the ARFF format used by WEKA. Each `FilterInstance` indicates which datatype it employs to process the data by using '*generic types*'. Based on the indicated type, the framework is able to automatically find the appropriate `Convertor` and `Deconvertor`. This eases the development of specific use cases and the usage of external libraries. The *Application Developers* only have to develop `Convertor` and `Deconvertor` subclasses that implement the conversion to the specific file format used by the FilterInstance.

To manage the complete pipeline, a special type of FilterInstance was created, namely the MiningManager. This class is responsible for constructing the *Learning Pipeline* out of the separate filters, starting it and processing the results. To implement a specific *Learning Pipeline*, a subclass of the MiningManager needs to be constructed that implements the setupPipeline method. This method initializes the different filters of the pipeline and connects them to each other. Each FilterInstance is connected to the next FilterInstance in the pipeline by using the connectNextFilter method. The first FilterInstance is connected to the MiningManager, while the last FilterInstance indicates the MiningManager as next filter to ensure proper processing of the result of the *Learning Pipeline*.

The ConfigurationModule is notified of which data should be collected for the mining process, either by the *Stakeholders* and *Application Developers* or by the *Monitoring Algorithms*. It configures the MiningManager to use the appropriate DataCollectionModule, InputConvertor and IntegrationModule that suits this type of data. It also passes the correct parameters to the DataCollectionModule, which are needed to retrieve the data from the *Persistency Layer*. Next, the ConfigurationModule calls the setupPipeline and startPipeline methods of the MiningManager to create the pipeline and start the learning process. The latter method first collects the necessary data by using the associated DataCollectionModule and converts it to the LearningDataObject format with the InputConvertor. Next, the MiningManager calls the process method of the first FilterInstance in the pipeline. This FilterInstance processes the data with its doProcessing method and then calls the process method of the next FilterInstance in the pipeline. This continues until the last FilterInstance calls the process method of the MiningManager. The MiningManger then finishes the learning process by calling the IntegrationModule to integrate the knowledge in the *Knowledge Base*.

It can be noted that the implemented framework is very extensible, modular and flexible, which allows easy adoption for any use case, as illustrated in the following section.

5.5 Use case: Mining the reasons for patients' call light use to automatically launch calls

5.5.1 Scenario description

Nurse call systems are a fundamental technology in continuous care as they are used by caregivers to coordinate work, be alerted of patients' needs, communicate with them through intercoms and request help from other staff members. When patients feel unwell they push a button. The nurses then receive a message with the room number on a beeper. This brings up the question: which nurse goes to the room? The closest one? the one on call, etc.? Current systems often have a very static nature as call buttons have fixed locations, e.g., on the wall next to the bed. There is an increased risk when patients become unwell inside a hallway, staircase or outside as they cannot use the nurse call system. Additionally, the current nurse call algorithms consist of predefined links between beeper numbers and rooms. Consequently, the system presently does not take into account the various factors specific to a given situation, such as the pathology of a patient, e.g., heart patient or confused, nor the competences of the staff, e.g., nurse or caregiver.

The increased introduction of electronic devices in continuous care settings facilitated the development of the ontology-based Nurse Call System (oNCS), which allows patients to walk around freely and use wireless nurse call buttons. Additionally, this platform manages the profiles of staff members and patients in an ontology. A sophisticated nurse call algorithm was developed by the authors. It first determines the priority of the call using probabilistic reasoning algorithms, which take into account the origin of the call and the pathology of the patient. Next, the algorithm finds the most appropriate staff member to handle the call. It dynamically adapts to the situation at hand by taking into account the context, e.g., location of the staff members and patients, the priority of the call and the competence of the different caregivers. The oNCS was implemented according to the *Context-Aware Platform* architecture discussed in Section 5.3 and visualized in Figure 5.1. A detailed description of this platform can be found in [44].

The oNCS is also able to automatically launch context calls based on the data generated by the electronic equipment and sensors in the environment, e.g., when a patient spikes a fever or when the light intensity is too high in the room of a patient with a concussion. It is however very difficult for developers to determine in advance all the risky situations for which a context call should be launched. These parameters and their thresholds are very dependent on the specific environment where the oNCS is deployed. Moreover, some of the relations between parameter measurements and calls made by the patient might not even be directly apparent to the caregivers as these relations are not rigorously studied.

To detect relations between the parameter measurements and the calls made by

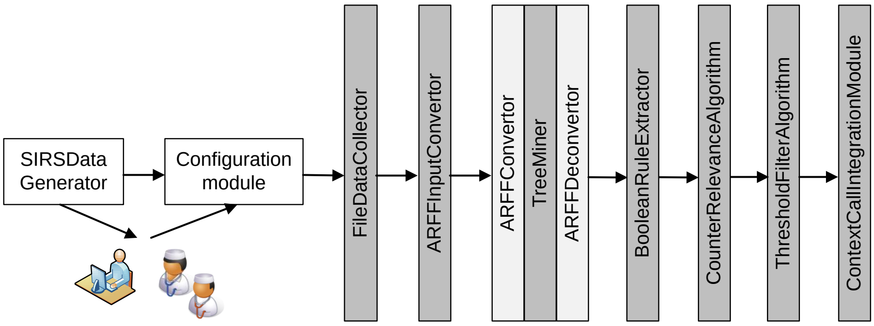


Figure 5.5: The pipeline used by the Learning Engine to tackle the SIRS use case

patients, the oNCS was extended with *Monitoring Algorithms*, the *Configuration Module* and *Learning Engine*. To evaluate this extension, a relation was simulated and it was investigated whether the *Learning Engine* was able to detect this trend and add it to the *Knowledge Base*. The trend that patients make a call when they exhibit symptoms for Systemic Inflammatory Response Syndrome (SIRS) [64, 65] was chosen as simulated relation. This medically relevant use case could be easily generated, but is challenging for the *Learning Engine* to detect. SIRS is a generalized inflammatory reaction of the organism to a severe medical condition such as acute pancreatitis, severe burn injury, trauma, surgical procedure or infection. If SIRS is the response to an infection, the patient is diagnosed with sepsis. Sepsis has a high mortality rate (30%-40%). The criteria for diagnosing a patient with SIRS are:

- Tachycardia: heart rate $>$ 90 beats per minute (bpm)
- Fever or hypothermia: body temperature $> 38\,^{\circ}\mathrm{C}$ or $< 36\,^{\circ}\mathrm{C}$
- Tachypnea: arterial partial pressure of carbon dioxide ($PaCO_2$) $<$ 32 mmHg
- White Blood Cell (WBC) count $<$ 4,000 cells/mm^3 or $>$ 12,000 cells/mm^3

For the diagnosis of SIRS, two or more of these criteria must be fulfilled. This is a challenging scenario for the *Learning Engine* as it involves both parameters measured at regular intervals by sensors, i.e., the heart rate and body temperature, as well as parameters obtained through the analysis of a blood sample by the laboratory, i.e., WBC and $PaCO_2$. Moreover, a combination of conditions needs to be fulfiled before the call should be launched.

The following sections illustrate how the *Learning Engine* was implemented and the *Learning Pipeline* was constructed, using the (abstract) classes discussed in Section 5.4, to detect this relation and add it to the *Knowledge Base*. The resulting pipeline is visualized in Figure 5.5.

Heart rate	**Body temperature**	**$PaCO_2$**	**WBC count**	**Call**
61.42	*38.62*	34.54	4969	No
78.55	37.47	32.68	7746	No
88.37	*35.76*	46.53	7253	Yes*
67.92	36.10	42.53	*12096*	Yes*
66.63	*40.95*	*30.56*	*3740*	Yes
91.59	36.78	*29.94*	*12301*	No*
94.52	*40.67*	*28.89*	4866	Yes
95.23	*35.93*	*31.61*	8737	No*

Table 5.1: Some example instances of the SIRS dataset

5.5.2 Scenario implementation

5.5.2.1 Generating the SIRS data

To realize the scenario, a dataset needs to be generated in which the trend can be detected that patients make calls when they exhibit SIRS symptoms. This dataset consists of a set of instances, each consisting of five data values, namely a value for the four SIRS parameters and whether or not a call was made. A *SIRS Instance* is defined as an instance, which consists of a combination of the four SIRS parameters that fulfills two or more SIRS criteria. Logically, a *Non-SIRS Instance* is defined as an instance, which fulfills at most one SIRS criterion at a time.

When the different instances are generated, each instance has 15% chance of being a *SIRS Instance*. The parameter values are randomly generated, while ensuring that at least two parameters fulfill the SIRS criteria for *SIRS Instances* and at most one criterion is fulfilled for *Non-SIRS Instances*. The values are generated within realistic bounds, e.g., temperature must be lower than $43\,^{\circ}\mathrm{C}$. Whether the *SIRS Instance* fulfills two, three or four criteria and whether the *Non-SIRS Instance* fulfills one criterion or none, is also randomly chosen.

Finally, each instance needs to be associated with a context call or not. To achieve a realistic dataset, noise is introduced by wrongly classifying the instances, i.e., associating *Non-SIRS Instances* with a call and vice versa. A noise percentage of x means that each *Non-SIRS Instance* has $x\%$ chance of being associated with a call and vice versa.

Some example instances are illustrated in Table 5.1. The first four instances are *Non-SIRS Instances*, while the latter four are *SIRS Instances*. The parameter values that fulfill SIRS criteria are indicated in italic. The calls marked with a *-symbol represent noise. A *Data Generator* was written to create the needed instances and provide them in the ARFF format, i.e., the data format used by WEKA. The resulting file is stored in the *Persistence Layer*.

5.5.2.2 The oNCS and continuous care ontologies

As mentioned in Section 5.2, little work has been done on the development of high-level ontologies, which can be used to model context information and knowledge utilized across the various continuous care settings, e.g., hospitals, homecare and residential care settings. Therefore we developed the *Continuous Care Ontology*, which models the generic context information gathered by the various sensors and devices, the different devices, the various staff members and patients and their profile information, medical conditions, roles and competences and the variety of tasks that need to be performed. A detailed description of this ontology can be found in [53]. The most important classes of these ontologies pertaining to the use case are visualized in Figure 5.6. This ontology references the *Galen Common Reference Model* [50] as *Medical Ontology*. The concepts from the *Galen Common Reference Model* are preceded by the `galen` namespace in Figure 5.6. The concepts preceded with the `temporal` namespace are imported from the *SWRL-TemporalOntology* [66] to represent temporal information. Finally, the *Wireless Sensor Network (WSN) Ontology* [67] was imported, as shown by the concepts preceded by the `wsn` namespace, to represent the knowledge pertaining to observations made by sensors. The *Probabilistic Domain Ontology* then models the specific properties of the environment where the oNCS is deployed, e.g., the specific roles and competences of the staff members and how they map on each other.

As can be seen, the model contains a `System` concept which models a system and its components. The ontology allows interpreting the data values monitored by the sensors. For this the ontology uses an observation pattern. A data value monitored by a system is modeled in the ontology as an `Observation`. Rules and axioms added to the ontology allow detecting specific phenomena in these observations, which are modeled as `Symptom` concepts. For example, the `TemperatureAbove38Symptom` class is defined as follows:

BodyTemperatureObservation AND $\exists$hasValue “ > 38”

This axiom ensures that a `BodyTemperatureObservation` of more than 38 °C is reclassified as a `TemperatureAbove38Symptom`. Similarly, symptoms can also be reclassified as faults and even as solutions and actions that should be taken.

People are modelled through the `Person` concept and their roles and competences can be indicated. It can also be indicated with which person the sensors are associated through the `associatedWith` relationship. The medical parameters collected about a patient, either by sensors, the observations of staff members or the analysis of blood samples, are modelled as `MedicalParameters`. Similar to observations, these parameters can also be reclassified as symptoms. The medical condition of a person can also be modeled, e.g., `Fever`.

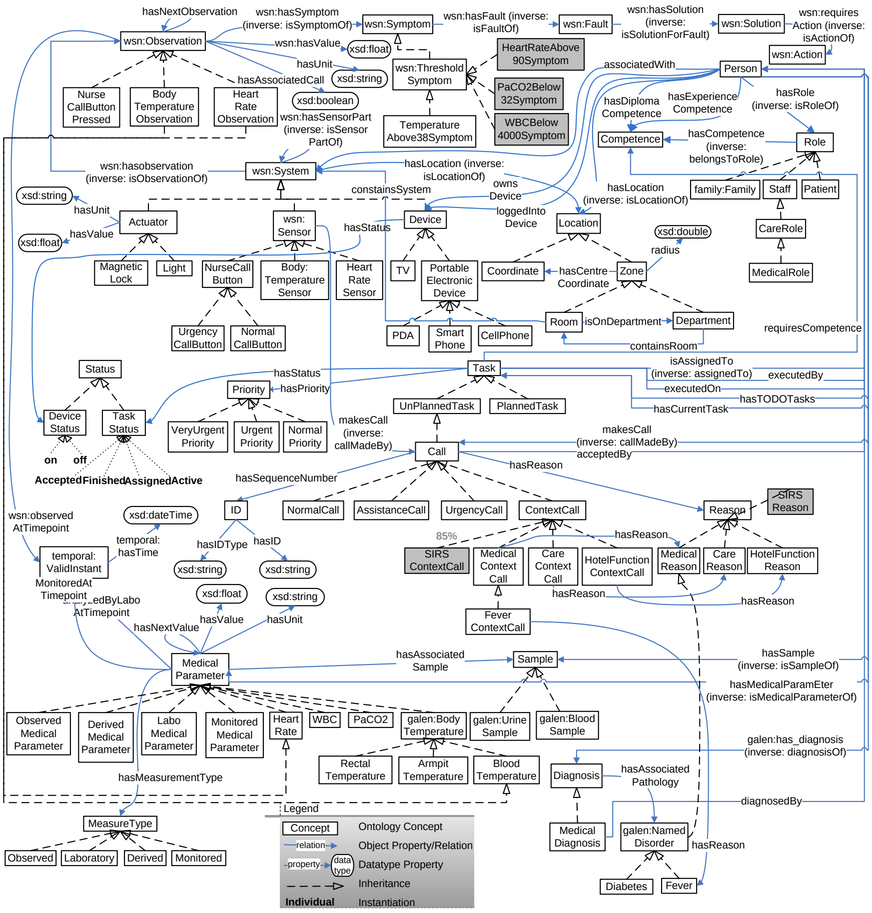


Figure 5.6: Prevalent classes of the Continuous Care, Medical and Probabilistic Domain ontologies pertaining to the SIRS use case

To model the daily activities of the caregivers and patients, the `Task` concept is used, which is further divided into planned and unplanned tasks. Each task can be assigned a `Status`, e.g., `Active` or `Finished`, a `Priority` and the competences which are needed to execute the task. People can be connected to the tasks through various relationships, e.g., `hasCurrentTask`, `isAssignedTo` or `executedBy`. A Call is modelled as an `unPlannedTask`. A call can be associated with a `Reason`, e.g., `Fever`. Four types of calls can be discerned. A `NormalCall` is a call made by a patient, while an `AssistanceCall` is launched by a caregiver to request help from another staff member. An `UrgencyCall` is only used for emergency situations, e.g., when a patient needs to be reanimated. Finally, a `ContextCall` is call that is automatically generated by the oNCS as a consequence of certain conditions being fulfilled. Consider for example the following Jena rule:

```
[FeverContextCall:
(?symp rdf:type oncs:TemperatureAbove38Symptom)
noValue(?symp task:hasAssociatedCall)
(?symp wsn:isObservationOf ?system)
(?kind rdf:type oncs:FeverContextCall)
→
createContextCall(?system, ?kind)
(?symp task:hasAssociatedCall 'true'^^xsd:boolean)]
```

The first line represents the name of the rule. First, it is sought if a body temperature of more than 38 °C was observed for which a call has not been launched yet. Next, the system that made the observation is retrieved. Finally, the type of call that should be created is specified. As a result, the functor `createContextCall` is called, which creates a `ContextCall` of type `FeverContextCall` and associates the system that made the observation with this call. The functor also assigns the status `Active` to the call. Moreover, the `hasAssociatedCall` relationship is set to `true` to make sure that the rule does not fire again.

The oNCS contains rules that fire when active calls are added to the ontology. Based on the context information, these rules assign the most appropriate staff member to the call. More information about these assignment rules can be found in [44].

Similar to how the fever example was modeled, the SIRS use case can be easily represented using these classes. Individuals of type `BodyTemperatureSensor` and `HeartRateSensor` are created to represent the sensors that measure the medical parameters of the patients. These sensors make observations of type `BodyTemperatureObservation` and `HeartRateObservation` respectively, which are associated with their sensors through the `hasObservation` relation. The measured value is indicated with the `hasValue` relation. Individuals of type `BloodSample` are created, that represent the blood samples analyzed by the laboratory to determine the WBC count and $PaCO_2$ of the patient. These results are captured in the ontology as medical parameters of type `WBC` and `PaCO2`. They are associated with their blood sample through the `hasAssociatedSample` relationship. Finally, when a patient makes a call by pressing a button, an individual of type `Call` is created in the ontology, which is connected through the `callMadeBy` relationship with the patient. Through reasoning, this call is reclassified as a `NormalCall` as it is made by a person with as role `Patient`.

A mobile nurse call application was also developed, which is used by the caregivers to receive, assess and accept, i.e., indicate that they are going to handle, calls. A nurse can also use the application to contact other staff members or the patient, e.g., to request the reason for the call or to give feedback. Before a nurse is able to indicate a call as finished, the reason for the call must be indicated either on the mobile application or the terminal next to the bed of the patient. This reason is also entered in the ontology. The mobile application is further explained in [68].

5.5.2.3 Collecting the data and input conversion

As the data is generated, no *Monitoring Algorithms* are needed. However, a *Monitoring Algorithm* could easily be written as follows. Relationships need to be found between medical parameters of patients and the calls that they make. The *Context Call Monitoring Algorithm*, monitors the ontology for calls of type `NormalCall`. When such a call is added to the ontology, the algorithm collects the most recent value for each medical parameter that is measured about the patient who made the call. This information can easily be retrieved using SPARQL queries. As not every medical parameter is measured for every patient, the dataset possibly contains missing values. When the call has been completely handled by the caregiver, the algorithm also retrieves the reason, which was attached to the call. As such, different data sets can be created, grouping calls together which have similar reasons. These datasets can differ in granularity of the reason. For example, a dataset could be created for all the calls with a `MedicalReason` or for all the calls with the more specific reason `Fever`. All calls of the second dataset would also be part of the first dataset, as `Fever` is a subclass of `MedicalReason`. Each of these datasets could be used as input for the *Learning Engine*. Other ways of grouping the data instances can also be employed, e.g., grouped per patient or grouping the instances of patients that have a similar pathology. The *Context Call Monitoring Algorithm* keeps track of how many instances have been collected for each dataset. When a representative amount has been gathered, the dataset is expanded with negative examples. For example, the medical parameters of the patients already present in the dataset can be collected at a timepoint when they have not seen a caregiver or made a call for a while or at a timepoint they made a call for a different reason. Finally, the *Monitoring Algorithms* invoke the *Configuration Module* to initiate the *Learning Engine*. The datasets can also be intermediately shown to the *Stakeholders* and *Application Developers* for inspection. In this use case, the *Data Generator* takes on the role of the *Monitoring Algorithm*.

The *Monitoring Algorithms* can store the datasets in the *Persistence Layer* in a format that best suits their needs. For the *Data Generator*, the ARFF format was chosen. Ontology individuals could also be directly stored in a triple store. The *Monitoring Algorithm* or the *Data Generator* indicates the location of the data and its format to the *Configuration Module*. They also indicate which `MiningMana-`

ger should be used to process the data. Different types of *Learning Pipelines*, which each consist of a combination of filters that suit the needs of a particular use case, can be created by implementing several subclasses of the MiningManager. This allows multiple *Monitoring Algorithms* to run at the same time and the collected data to be processed by the MiningManager, and thus *Learning Pipeline*, that matches with the goal of the algorithm.

The *Configuration Manager* configures the MiningManager to use the appropriate DataCollectionModule and InputConvertor that suits the format of the data. The subclass FileDataCollector of the DataCollectionModule class was implemented, which is able to read the data from a file at a specified location. The result is a String, which is provided to the appropriate ARFFInputConvertor. This subclass of InputConvertor is able to translate this ARFF-String to the LearningDataObject format, which is used by the *Learning Pipeline*. During the translation it also checks if the specified value for an attribute, e.g., 38 for the body temperature parameter, is compatible with the type of this attribute. For example, it is not allowed to assign a String to a numerical attribute. Illegal data instances are discarded.

5.5.2.4 Mining the sensor data using a C4.5 decision tree

A *Pre-Processing* filter was not implemented for this use case, as it works on generated data. If the previously discussed *Context Call Monitoring Algorithm* was used, several *Pre-Processing* filters could be used. For example, a RemoveOutliers filter could be employed to remove outliers or impossible parameter values in the dataset. Moreover, the number of features, i.e., measured medical parameters, in the dataset would be relatively high. A FeatureSelection filter could be used to select the most interesting features for the *Data Mining* step. Finally, a MissingValues filter would be able to deal with the missing values in the dataset.

The *Data Mining* filter needs to find relations in the generated dataset between the sensor measurements and the occurence of a call. Supervised [56] classification techniques [69] consider a set of input attributes, e.g., the different sensor types, and a output attribute, also called the class attribute or the label, e.g., whether a call was made or not. These techniques then try to build a model that fits this data set and derives relationships between the input attributes and the label. Building a decision tree [70] based on the information captured in the dataset is a well-known and easy supervised classification technique. A decision tree is a tree structure in which each leaf represents a value that the label can assume, e.g., Yes or No. The internal nodes of the tree represent the attributes on which a decision is based, while the branches represent conditions that the attributes need to fulfill. As an example, a part of the decision tree of the SIRS example is shown in Figure 5.7. To determine the label of a certain data instance, one just needs to follow the tree

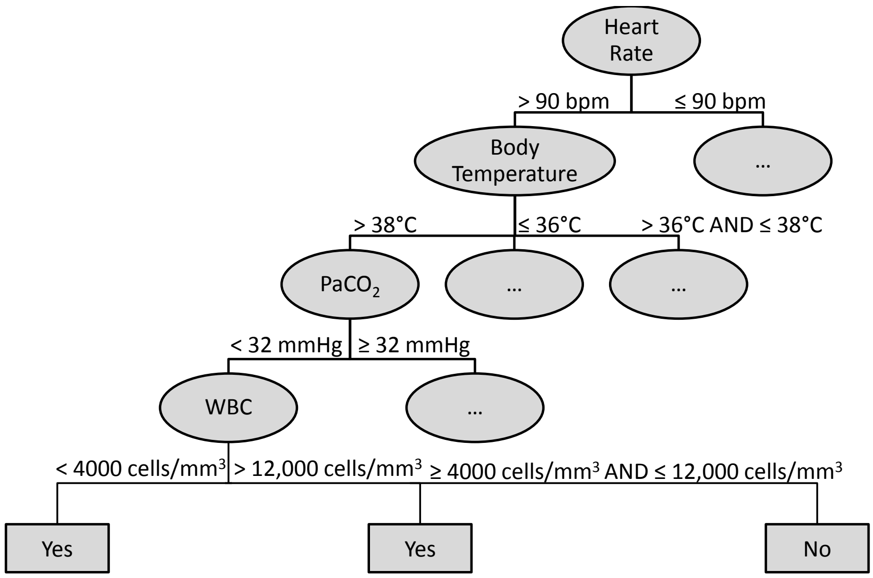


Figure 5.7: Part of the decision tree of the SIRS example

from the root to the leaves along the branches for which the instance fulfills the conditions. Essentially, a decision tree forms a compact representation of classification rules. For example, the decision tree shown in Figure 5.7 contains the classification rule:

$$\text{HeartRate} > 90 \text{ bpm AND BodyTemperature} > 38\,^{\circ}\text{C AND} \\ < 32 \text{ mmHg AND} < 4000 \text{ cells/mm}^3 \rightarrow \text{Yes}$$

Different techniques can be used to build such a decision tree out of a data set, e.g., the Iterative Dichotomiser 3 (ID3) [71] or C4.5 [72] algorithm. The latter is a more sophisticated algorithm as it allows that attributes have numeric values [73], is able to handle missing values and prunes the tree in order to make it more compact and avoid overfitting [74].

To implement the C4.5 decision tree, an external library is used, namely WEKA. WEKA provides its own implementation of the C4.5 algorithm, namely J4.8, which was used in this research. A subclass of the `FilterInstance` abstract class was implemented, called `TreeMiner`. As previously mentioned, WEKA uses the ARFF data format to represent data. Therefore, an `ARFFDataObject` was created as a subclass of `DataObject` and (de)convertors were implemented that are able to translate the internal data format of the *Learning Engine*, i.e., `LearningDataObject`, to and from the ARFF data format. As mentioned in Section 5.4, it is enough to indicate in the `TreeMiner` that the filter uses the

ARFFDataObject in the getDataType method and the framework will automatically use the correct (de)convertors to transform the data. Which attribute should be used as label can be indicated in the TreeMiner class. In case the label is not indicated, the TreeMiner assumes that the last attribute in the data format is the label. The ARFFInputConvertor, discussed in the previous section, makes sure that the last attribute is indeed the label. The doProcessing method then calls the Java API of WEKA to build the decision tree. However, the J4.8 algorithm does not allow to retrieve separates branches and nodes of the tree. Only a textual representation of the complete decision tree can be obtained. For example, the textual representation of the tree visualized in Figure 5.7 is:

N0 [label="HeartRate"]

N0 $\rightarrow$ N1 [label=" $>$ 90"]

N1 [label="BodyTemperature"]

N1 $\rightarrow$ N2 [label=" $>$ 38"]

N2 [label="PaCO2"]

N2 $\rightarrow$ N3 [label=" $<$ 32"]

N3 [label="WBC"]

N3 $\rightarrow$ N4 [label=" $<$ 4000"]

N4 [label="Yes"]

The nodes and branches are identified and translated to the LearningDataObject format such that the result can be forwarded to the next step in the pipeline. It is important to note that new results are always added to the data being exchanged, so that the original data set also stays available for the following steps in the pipeline.

The *Post-Processing* filter is responsible for deriving the rules out of the textual representation of the decision tree provided by the J4.8 algorithm. Therefore, the BooleanRuleExtractor subclass of the FilterInstance class was implemented. The implemented doProcessing method takes into account that the label has a boolean value, i.e., Yes or No. Only the branches that result in a positive leaf need to be translated into a rule, as only those rules will result in calls. The doProcessing method starts from a positive leaf and follows the branches until the root is reached. Each branch that is crossed is added as a condition in the rule. The iterative build-up of the rule according to the output of the J4.8 algorithm illustrated in Figure 5.7 is as follows:

Step 1: $\rightarrow Yes$

Step 2: WBC $<$ 4000 $\rightarrow$ Yes

Step 3: PaCO2 $<$ 32 AND WBC $<$ 4000 $\rightarrow$ Yes

Step 4: BodyTemperature $>$ 38 AND PaCO2 $<$ 32 AND
WBC $<$ 4000 $\rightarrow$ Yes

Step 5: HeartRate $>$ 90 AND BodyTemperature $>$ 38 AND
PaCO2 $<$ 32 AND WBC $<$ 4000 $\rightarrow$ Yes

The resulting rules are represented in the `LearningDataObject` format such that they can be processed by the *Decision Module*.

5.5.2.5 Filtering and integrating the rules

As mentioned in Section 5.3, probabilities are attached to the discovered rules to express their reliability to the users and to ensure that the *Knowledge Base* remains consistent, i.e., that the new knowledge does not contradict already existing knowledge.

To calculate the initial probability, the `CounterRelevanceAlgorithm` was implemented as a subclass of the `FilterInstance` class. This algorithm applies the rule to the original dataset, which is still included in the `LearningDataObject`. The percentage of times that the rule labels the data correctly, i.e., the conditions of the rule are fulfilled and the label is Yes, is used as probabilistic value. As the data for this use case was generated, this probability thus reflects the amount of noise in the dataset. For the remainder of the text, it assumed that the rule, which was presented in the previous section, receives a probability of 85%.

A simple filter algorithm, namely the `ThresholdFilterAlgorithm` was implemented as subclass of the `FilterInstance` class. This algorithm filters the rules for which the probability is lower than a specified probability, e.g., 50%. This rule is thus not added to the *Knowledge Base*. However, the rule and its associated probability is archived in the *Persistence Layer*.

Finally, the `ContextCallIntegrationModule`, a subclass of the `IntegrationModule` class, is responsible for integrating the rules and associated probabilities in the *Knowledge Base*. First, new subclasses of `ContextCall` and `Reason` are introduced in the ontology, with as name the condition of the rule added before the suffix `ContextCall` and `Reason` respectively. For brevity, `SIRSContextCall` and `SIRSReason` are used to refer to the concepts that are created for the rule, which is used as running example, i.e., the rule that fulfills each of the four criteria. These concepts are visualized in grey in Figure 5.6.

Pronto is used to represent and reason on the probabilistic information in the ontology. To express generic probabilistic knowledge, Pronto uses Generic Conditional Constraints (GCCs) [75]. Generic means that the knowledge does not apply to any specific individual but rather to a fresh, randomly chosen one. A GCC is of the form (D—C)[l,u] where D and C are classes in the ontology and [l,u] is a closed subinterval of [0,1]. To represent these GCCs in the ontology, Pronto employs subsumption axiom annotations. For example, to express the fact that the `SIRSContextCall` is a `ContextCall` with only 85% probability, the following subsumption axiom annotation is added to the ontology:

< owl11:Axiom >
< rdf:subject rdf:resource="#SIRSContextCall" >
< rdf:predicate rdf:resource="&rdfs;subClassOf" >
< rdf:object rdf:resource="#ContextCall" >
< pronto:certainty > 0.85;1 < /pronto:certainty >
< owl11:Axiom >

Second, a `Symptom` concept is created for each parameter condition in the discovered rule, for example `HeartRateAbove90Symptom`, `BodyTemperatureAbove38Symptom`, `PaCO2Below32Symptom` and `WBCBelow4000 Symptom`. These classes are defined by axioms, for example the `HeartRateAbove90Symptom` is defined as:

HeartRateObservation AND ∃hasValue " $>$ 90"

If a class with a similar definition already exists, the existing class is used. This can be checked by searching for equivalent classes in the ontology with a Reasoner. For example, `BodyTemperatureAbove38Symptom` is not added to the ontology, as `TemperatureAbove38Symptom` is an equivalent class. The newly created `Symptom` classes are visualized in grey in Figure 5.6.

Third, the rules are translated to a Jena Rule using the created classes and added to the *Knowledge Base*. For example, the rule from the previous section is translated to four Jena Rules. For example, the following Jena Rule launches when all the requirements are met and at least one of the symptoms does not have an associated call yet:

[SIRSContextCall:
(?symp1 rdf:type oncs:HeartRateAbove90Symptom)
noValue(?symp1 wsn:hasNextObservation)
(?symp1 wsn:isObservationOf ?system1)
(?system1 sensor:associatedWith ?pat)
(?symp2 rdf:type oncs:TemperatureAbove38Symptom)
noValue(?symp2 wsn:hasNextObservation)
(?symp2 wsn:isObservationOf ?system2)
(?system2 sensor:associatedWith ?pat)
(?symp3 rdf:type oncs:PaCO2Below32Symptom)
noValue(?symp3 medical:hasNextValue)
(?symp3 wsn:isObservationOf ?system3)
(?system3 sensor:associatedWith ?pat)
(?symp4 rdf:type oncs:WBCBelow4000Symptom)
noValue(?symp4 medical:hasNextValue)
(?symp4 wsn:isObservationOf ?system4)
(?system4 sensor:associatedWith ?pat)
noValue(?symp1 task:hasAssociatedCall)
(?kind rdf:type oncs:SIRSContextCall)
→
createContextCall(?system1, ?kind)
(?symp1 task:hasAssociatedCall 'true'^^xsd:boolean)]
(?symp2 task:hasAssociatedCall 'true'^^xsd:boolean)]
(?symp3 task:hasAssociatedCall 'true'^^xsd:boolean)]
(?symp4 task:hasAssociatedCall 'true'^^xsd:boolean)]

For each symptom a rule is created. The only difference between the rules is that the condition for an associated call is checked for a different symptom each time. This is because the different symptoms on their own might already have launched context calls for other reasons, e.g., the `TemperatureAbove38Symptom` might already have launched a `FeverContextCall`. Afterwards, all the symptoms are associated with a call to ensure that only one context call is launched. The rule also ensures that the most recent parameter values are taken into account by checking whether there are no next observations or parameter values through the `hasNextObservation` and `hasNextValue` relations.

When the rule is fulfilled, a new context call is added to the *Knowledge Base*. Consequently, the oNCS will detect the new context call and assign a staff member to it. The Pronto reasoner can then be used to retrieve the probabilistic information associated with the call. This information can then be conveyed to the assigned caregiver through the mobile application.

As only subclasses are added to the ontology and no knowledge is removed, it is unlikely that the ontology will become inconsistent. However, if the ontology does become inconsistent, the following solution can be employed. When new information is added to the ontology, the consistency is checked. If the ontology is no longer consistent, the information is identified with which the new knowledge conflicts. Pronto allows that different chunks of probabilistic information conflict with each other. For example, a bird is flying object with high probability and all penguins are birds, but a penguin has a low probability of flying. More specific probabilistic constraints are thus allowed to override more generic ones. The conflicting information is annotated with the probabilistic interval [1,1], which indicates that the knowledge is generally true. Consequently, we are now dealing with conflicting, probabilistic knowledge and the rule of increasing specificity can be employed to resolve the conflict. As such, we ensure that the ontology remains consistent.

Finally, the *Integration Module* also associates the learned knowledge with information about the *Learning Engine* that created it by using concepts from the *Learning Ontology*. The individuals, which are created to realize this goal, are visualized in bold in Figure 5.2.

Note that `ContextCall`, `Symptom` and `Reason` concepts and an associated probabilistic annotation axiom and Jena Rule are created for each discovered rule.

5.5.2.6 Adapting the probabilities

This step was not implemented as it requires the system to be deployed such that information about the usage of the new knowledge by the caregivers can be acquired. However, it is briefly discussed how this task of adapting the probabilities could be realized for this use case.

A *Monitoring Algorithm* could be implemented, which takes as parameter the newly created context call, e.g., in this case `SIRSContextCall`. The algorithm monitors the *Knowledge Base* and collects calls of this type, which have been launched by the system. For each call, its reason and the symptoms that caused the calls to be launched are retrieved. When nurses handle calls, they need to input the reason for the call. For context calls, they can affirm the reason, which was assigned by the framework, e.g., SIRS. They can also choose to change it, e.g., to false because the call was unnecessary. As such a dataset is created for each rule, which maps the values of the medical parameters on the associated reason. When a representative amount of data has been collected, this dataset can be retrieved by the `FileDataCollector` and converted by the `ARFFInputConvertor`. The output can then be processed by a *Learning Pipeline* consisting of only one filter. This filter is a *Probabilistic Relevance Algorithm*, which simply calculates the percentage of calls for each rule for which the reason was not changed. This means that caregivers deemed the reason to be correct. This percentage is than given to the *Integration Module*, which adapts the probability for this rule in the ontology to this calculated percentage. As explained in Section 5.3.3.3, if the calculated percentage exceeds or falls below the probability thresholds specified in the *Integration Module*, the knowledge is removed from the ontology or added as generally accepted knowledge without a probability.

5.5.3 Evaluation set-up

To evaluate the applicability of the framework, it is important to assess the correctness of the derived rules. The correctness of the used data mining techniques is influenced by the size of the dataset and the amount of noise. To assess the influence of the latter, the *Learning Pipeline* was consecutively applied to datasets of the same size, but with an increasing amount of noise. The amount of noise is varied from 0% to 50% in steps of 1%. As mentioned in Section 5.5.2.1, a noise percentage of x means that each *Non-SIRS Instance* has x% chance of being associated with a call and vice versa. It is unnecessary to increase the noise percentage beyond 50% as a random label is assigned at this point and the dataset becomes meaningless. The amount of noise needs to be increased in a dataset of realistic size. The WBC and $PaCO_2$ parameters are derived by the laboratory by analyzing a blood sample. Consequently, it is unlikely that more than two different values for these parameters will be generated per patient per day. If we assume that a department contains on average 30 patients and that we want to wait at most 28 days before we run the self-learning framework for the first time, a realistic dataset contains 1,680 instances, i.e., 30 patients x 28 days x 2 entries per patient per day.

The influence of the size of the dataset on the correctness is evaluated by consecutively applying the *Learning Pipeline* to datasets of increasing size. The dataset sizes range from 100 to 2,000 instances in steps of 100 instances. It can be noted that this range also contains the size of the dataset used for the correctness tests that evaluate the influence of noise, i.e., 1,680 instances.

It is also important to evaluate the performance, i.e., execution time and memory usage, of the developed *Learning Engine*. Although, the learning process will mostly run in the background, it is important to assess the amount of resource usage. Most healthcare environments have a limited amount of resources and delegating the processing to the cloud is often difficult because of privacy issues. To evaluate the influence of noise on the performance, the same datasets were used as for the correctness tests. However, to assess the influence of the size of the dataset, datasets were generated with sizes ranging from 1,000 to 30,000 in steps of 1,000 instances. Bigger datasets were used as it is important to explore the limits of the proposed framework.

To achieve reliable results, each test was repeated 35 times, of which the first three and the last two were omitted during processing. For each run, a new dataset was generated. Finally, the averages across the 30 remaining runs are calculated and visualized in the form of graphs. The tests were performed on a computer with the following specifications: 4096 megabyte (MB) (2 x 2048 MB) 1067 megahertz (MHz) Double Data Rate Type Three Synchronous Dynamic Random Access Memory (DDR3 SDRAM) and an Intel Core i5-430 Central Processing Unit (CPU) (2 cores, 4 threads, 2.26 gigahertz (GHz), 3 MB cache).

The term detection rate is introduced to assess the correctness. The SIRS use case is detected when the criteria for each of the four parameters of the SIRS use case are discovered. If one or more criteria is not learned, the SIRS use case is considered undetected. The detection rate of a dataset with a particular size is defined as the percentage of the 30 test runs for this size for which the SIRS use case was completely detected. For example, a detection rate of 50% for a dataset of 100 instances means that for 15 test runs of this dataset size the SIRS criteria were detected.

To assess the correctness, the relative error of the SIRS criteria is calculated. The relative error expresses how much the learned criterion deviates from the actual SIRS criterion. For example, a relative error of 5% for the "*heart rate > 90*" criterion indicates that the discovered threshold deviates from 90 by 5%. Note that the body temperature and WBC parameters have both an upper and lower threshold, while the heart rate and $PaCO_2$ have only one threshold.

5.5.4 Results

5.5.4.1 Correctness

Figure 5.8 depicts the detection rate as a function of the size of the dataset. The detection rate is relatively low for small datasets, but it quickly increases and reaches 100% for a dataset with 800 instances. When a dataset contains at least 1,000 instances, the detection rate is always 100%.

The detection rate is off course related to the relative error. In Figure 5.9 the relative error is depicted for each of the SIRS criteria as a function of the size of the dataset. A missing value, i.e., the criterion was not learned, corresponds to a relative error of 100%. Consequently, a low detection rate corresponds to high relative error. When the dataset reaches a 1,000 instances and a detection rate of

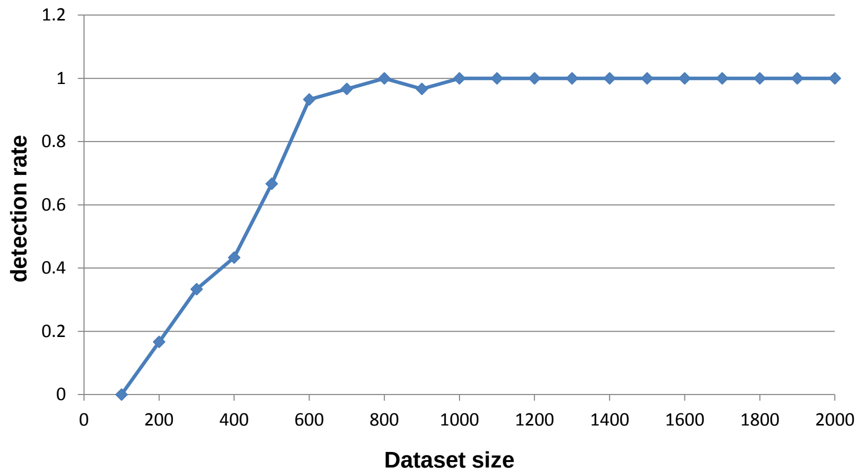


Figure 5.8: The detection rate of the SIRS use case as a function of the size of the dataset

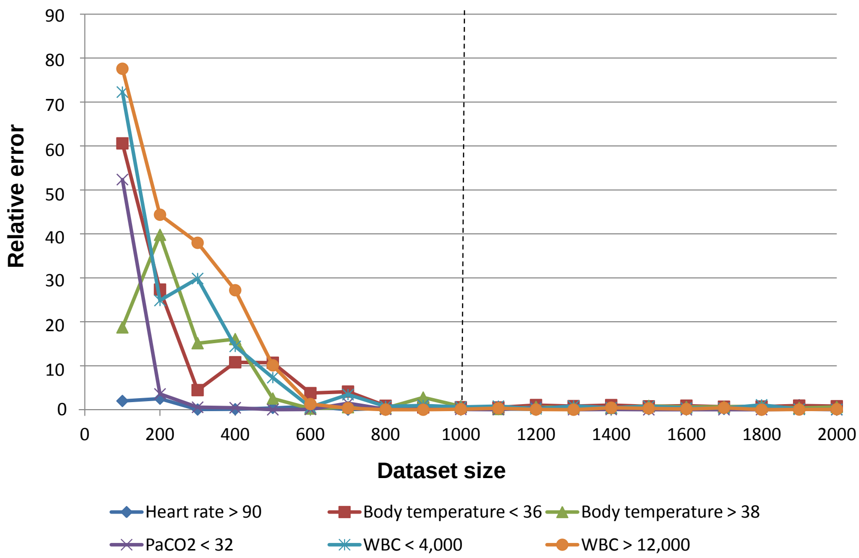


Figure 5.9: The relative errors for each of the SIRS criteria as a function of the size of the dataset

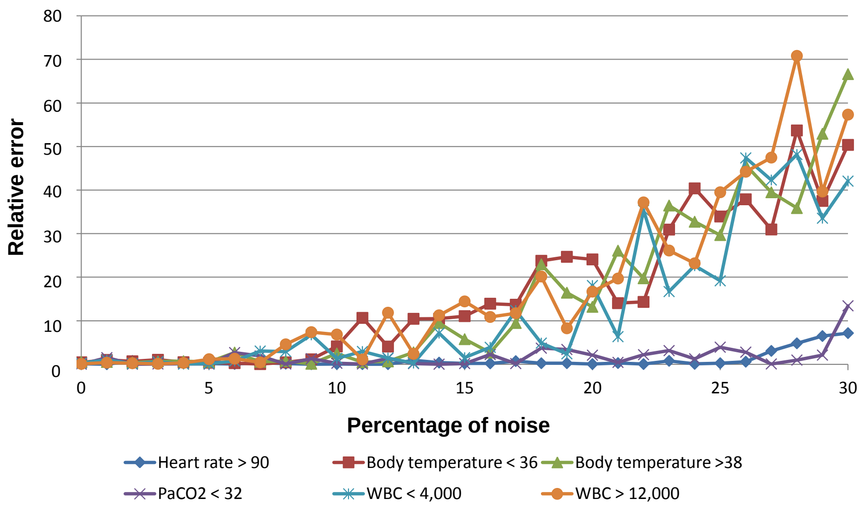


Figure 5.10: The relative errors for each of the SIRS criteria as a function of the amount of noise in the dataset

100% is thus achieved, the relative error stays below 1%. This means that for a dataset of 1,000 instances, the threshold is discovered for each criterion and it only deviates from the actual threshold by at most 1%, which is a very good result. If we consider that the parameters are collected twice a day for each patient in a department with 30 patients, enough instances would be collected after 17 days.

Figure 5.10 visualizes the relative errors for each of the SIRS criteria as a function of the amount of noise in a realisticly sized dataset of 1,680 instances. It is clear that the *Learning Engine* is insensitive to a noise rate of less than 5%. If the amount of noise increases, the relative errors quickly rise to 10% and higher. A relative error of 10% on the lower threshold of the body temperature, already implies a difference of 3.6 °C. This is unacceptable. In contrast, a relative error of 10% on the lower bound of the WBC only indicates a difference of 400 cells/mm^3. The acceptability of the relative error thus depends on the kind and range of the parameter.

5.5.4.2 Performance

The execution time as a function of the size of the dataset is depicted in Figure 5.11. Only the execution time of the most relevant pipeline steps is shown. The execution time of the `CounterRelevanceAlgorithm` and `ThresholdFilterAlgorithm` is negligible compared to the execution times of the visualized modules. The execution time of the `ContextCallIntegration-`

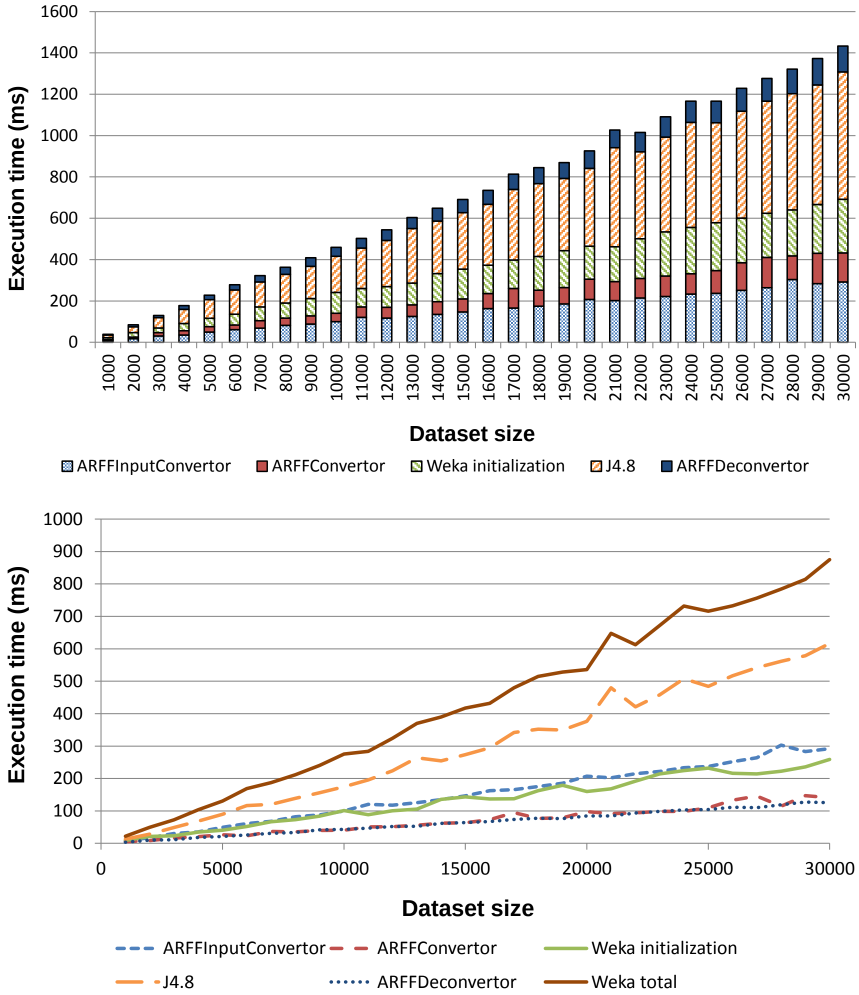


Figure 5.11: Execution time as a function of the dataset size

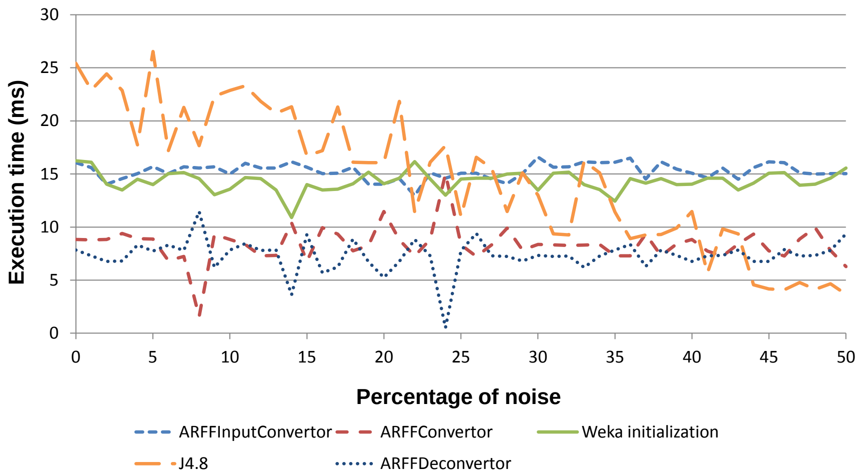
`Module` depends heavily on the complexity and the amount of data in the ontology. As the ontology was not initialized with a realistic data set, e.g., representing a realistic amount of staff members and patients, the execution time of this module is not shown. The size of the decision tree build by WEKA depends on the number of attributes in the dataset, but is independent of the number of instances, i.e., the size of the dataset. As the number of attributes, namely the four SIRS parameters and the label, stays constant and the Post-Processing step only processes the model build by WEKA, the execution time of this step is not influenced by the size of the dataset. Moreover, the execution time of the `BooleanRuleExtractor` was also negligible compared to the execution times of the depicted modules. The processing of the data by WEKA can be split up into two steps, namely transforming the ARFF format to Java Objects and the actual execution of the J4.8 algorithm to build the model. The execution times of both these steps are visualized.

It can be derived from Figure 5.11a that the execution time of the self-learning framework is linear as a function of the size of the dataset. The execution of the J4.8 algorithm by WEKA consumes the largest amount of execution time. It can also be noted that the `ARFFInputConvertor` consumes a considerable amount of execution time. This *InputConvertor* needs to translate a `String`-based representation of an ARFF-file to the internal data format used by the *Learning Pipeline*, namely `LearningDataObject`. Moreover, it needs to check if each value also fulfills the type requirements of the attribute, e.g., that a `String` is not provided where a numerical value is expected. The `ARFFConvertor` and `ARFFDeconvertor`, which are used by the Data Mining step to translate the internal data format to and from the ARFF format used by WEKA, are more performant. This is because these convertors translate to and from a Java Object representation of the internal format, which is more structured and is thus processed more easily.

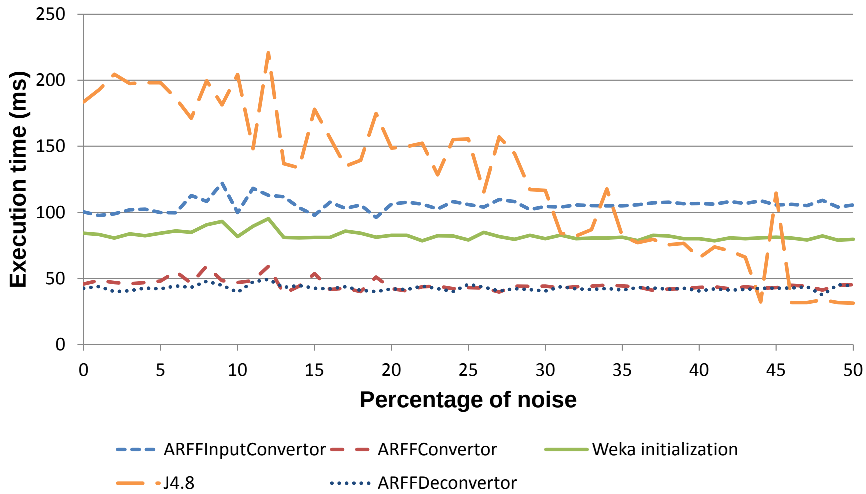
Figure 5.11b illustrates that the execution time of each of the visualized modules is also linear as a function of the size of the dataset. The complexity of the J4.8 algorithm is $O(m * n^2)$ for a dataset with m instances and n attributes [76]. Since the number of attributes is constant in this use case, this reduces to a complexity, which is linear in the number of instances, i.e., $O(m)$. The `ARFFInputConvertor`, `ARFFConvertor` and `ARFFDeconvertor` are also linear in the size of the dataset, as they need to (de)convert all the instances in the dataset one by one.

The execution time needed to process the dataset of realistic size, i.e., 1680 instances, is lower than 100 ms, which is a negligible delay. This means that the monthly patient data of a department with on average 30 patients can be processed very efficiently.

Figure 5.12a depicts the execution time as a function of the amount of noise for the realistic dataset containing 1680 instances. As the measured execution times are quite small, i.e., lower than 30 ms, the graphs are quite erratic and unpredictable. To get a clear view on the underlying trends, the performance tests were repeated for a dataset consisting of 5,000 instances. The amount of noise in this dataset is also gradually increased. The resulting graph is visualized in Figure 5.12b. It can be noted that only the execution time of the J4.8 algorithm is influenced by the amount of noise in the dataset. The execution time of the J4.8 algorithm decreases as the amount of noise in the dataset increases. It can be derived that the execution time decreases faster when the percentage of noise is higher than 5%. As shown in the previous section, the relative error quickly increases once the amount of noise rises above 5%. This is because the J4.8 algorithm will more



(a) Dataset of 1680 instances



(b) Dataset of 5000 instances

Figure 5.12: Execution time as a function of the amount of noise in the dataset

quickly decide that it is no longer useful to try to split up the decision tree. On the one hand, this leads to a lower detection rate as not all the criteria are discovered. On the other hand, this decreases the needed execution time of the algorithm.

Figure 5.13 illustrates the memory usage of the framework as a function of the size of the dataset. The fluctuating pattern of the graphs can be explained by the memory that is consumed by the *Garbage Collector* in Java. However, trend lines can clearly be discerned. It can be noted that the memory usage is linear as

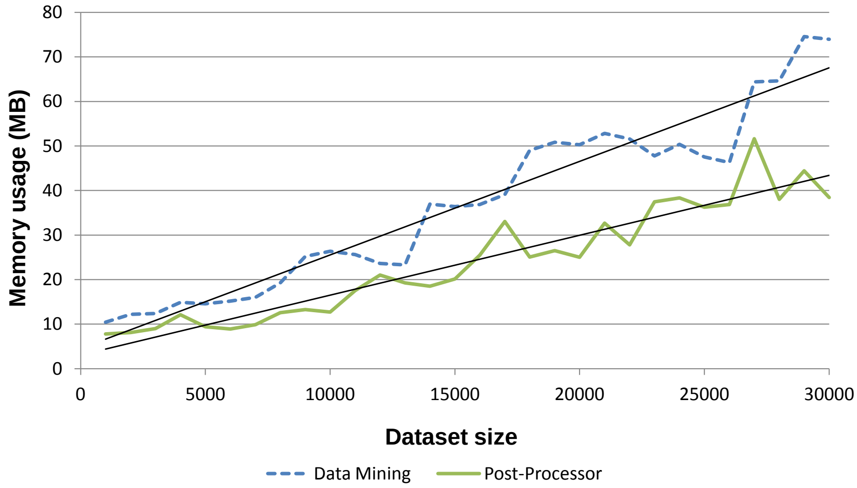


Figure 5.13: The memory usage as a function of the size of the dataset

a function of the amount of instances. Moreover, the total amount of consumed memory stays quite low, i.e., at most 80 MB. For the realistic dataset of 1680 instances the memory usage is negligible, namely about 10 MB.

It can be concluded that a dataset of realistic size for the SIRS use case can be processed by any modern PC or server and no cloud-based solutions are needed to run the framework.

5.6 Conclusions

In this paper a self-learning, probabilistic, ontology-based framework was presented, which allows context-aware applications to adapt their behavior at runtime. The proposed framework consists of the following steps. First, an ontology-based context model with accompanying rule-based context-aware algorithms is used to capture the behavior of the user and the context in which it is exhibited. Historical information is then gathered by algorithms that identify missing or inaccurate knowledge in the context-aware platform. This historical information is filtered, cleaned and structured so that it can be used as input for data mining techniques. The results of these data mining techniques are then prioritized and filtered by associating probabilities, which express how reliable or accurate they are. These results and the associated probabilities are then integrated into the context model and dynamic algorithms. These probabilities clarify to the stakeholders that this new knowledge has not been confirmed by rigorous evaluation. Finally, these probabilities are adapted, i.e., in- or decreased, according to context and behavioral information gathered about the usage of the learned information.

The pipeline architecture of the framework was presented and its implementation was detailed. Finally, a representative use case was presented to illustrate the applicability of the framework, namely mining the reasons for patients' nurse call light use to automatically launch calls. More specifically, detecting SIRS as a reason for nurse calls was used as a realistic scenario to evaluate the correctness and performance of the proposed framework. It is shown that correct results are achieved when the dataset contains at least 1000 instances and the amount of noise is lower than 5%. The execution time and memory usage are also negligible for a realistic dataset, i.e., below 100 ms and 10 MB.

Future work will mainly focus on the development of more intricate monitoring, probabilistic relevance and filter algorithms. Moreover, a prototype of the proposed framework will be deployed and evaluated in a real-life setting.

Acknowledgements

We would like to acknowledge that part of this research was supported by the iMinds Project ACCIO co-funded by the IWT, iMinds and the following partners: Televic NV, Boone NV, Dominiek Savio Instituut and In-Ham. F. Ongenae and M. Claeys would like to thank the IWT for financial support through their Ph.D. grant. The authors thank F. De Backere for support in constructing the figures.

References

[1] C. Orwat, A. Graefe, and T. Faulwasser. *Towards pervasive computing in health care - A literature review*. BMC Medical Informatics and Decsion Making, 8(26):18, 2008.

[2] K. Colpaert, S. V. Belleghem, D. Benoit, K. Steurbaut, F. D. Turck, and J. Decruyenaere. *Has information technology finally been adopted in intensive care units?* In 22nd Annual Congress of the European Society of Intensive Care Medicine, page 235, Vienna, Austria, October 11-14 2009.

[3] M. Tentori, D. Segura, and J. Favela. *Chapter VIII: Monitoring hospital patients using ambient displays*. In P. Olla and J. Tan, editors, Mobile Health Solutions for Biomedical Applications, volume 1, pages 143–158. Medical Information Science Reference;, Hershey, New York, USA, 1st edition, 2009.

[4] J.-C. Burgelman and Y. Punie. *Close encounters of a different kind: ambient intelligence in Europe*. True Vision: The Emergence of Ambient Intelligence, pages 19–35, 2006.

[5] T. Chin. *Technology Valued, but Implementing it into Practice is Slow*. American Medical News, 2004. http://www.ama-assn.org/amednews/2004/01/19/bisb0119.htm.

[6] J. Anderston and C. Aydin. *Evaluating the Impact of Health Care Information Systems*. International Journal Technology Assessment in Health Care, 13(2):380–393, 1997.

[7] J. H. Jahnke, Y. Bychkov, D. Dahlem, and L. Kawasme. *Context-aware information services for health care*. In Proc. of the Workshop on Modeling and Retrieval of Context, pages 73–84, 2004.

[8] J. Criel and L. Claeys. *A transdisciplinary study design on context-aware applications and environments. A critical view on user participation within calm computing*. Observatorio, 2(2):57–77, 2008.

[9] A. K. Dey and G. D. Abowd. *Towards a better understanding of context and context-awareness*. In D. R. Morse and A. K. Dey, editors, Proceedings of the CHI Workskhop on the What, Who, Where, When and How of Context-Awareness, pages 304–307, The Hague, The Netherlands, 1-6 April 2000. New York, NY, USA: ACM Press.

[10] H. Byun and K. Cheverst. *Utilizing context history to provide dynamic adaptations*. Applied Artificial Intelligence, 18(6):533–548, 2004.

[11] J. Hong, E. Suh, and S. Kim. *Context-aware systems: A literature review and classification*. Expert Systems with Applications, 36(4):8509–8522, 2009.

[12] M. Baldauf, S. Dustdar, and F. Rosenberg. *A survey on context-aware systems*. International Journal of Ad Hoc and Ubiquitous Computing, 2(4):263–277, 2007.

[13] W. Xue and H. K. Pung. *Chapter 8: Context-Aware Middleware for Supporting Mobile Applications and Services*. In A. Kamur and B. Xie, editors, Handbook of Mobile Systems Applications and Services, volume 1, pages 269–304. CRC Press;, Florida, USA, 1st edition, 2012.

[14] O. Yilmaz and R. C. Erdur. *iConAwa - An intelligent context-aware system*. Expert Systems with Applications, 39(3):2907–2918, 2012.

[15] T. Strang and C. Linnhoff-Popien. *A context modeling survey*. In J. Indulska and D. D. Roure, editors, Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp), Workshop on Advanced Context Modelling, Reasoning and Management, pages 31–41, Nottingham, UK, 7 September 2004.

[16] T. Gu, H. K. Pung, and D. Q. Zhang. *A service-oriented middleware for building context-aware services*. Journal of Network and Computer Applications (JNCA), 28(1):1–18, 2005.

[17] H. Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, University of Maryland, Baltimore County, 2004.

[18] L. O. B. S. Santos, R. P. Wijnen, and P. Vink. *A service-oriented middleware for context-aware applications*. In Proceedings of the 5th International Workshop on Middleware for Pervasive and Ad hoc Computing, pages 37–42, Newport Beach, Orange County, CA, USA, 26-30 November 2007. New York, NY, USA: ACM Press;.

[19] M. Román, C. Hess, R. Cerqueira, R. H. Campbell, and K. Nahrstedt. *Gaia: A Middleware Infrastructure to Enable Active Spaces*. IEEE Pervasive Computing, 1:74–83, 2002.

[20] S. L. Tsang and S. Clarke. *Mining User Models for Effective Adaptation of Context-aware Applications*. International Journal of Security and its Applications, 2(1):53–62, 2008.

[21] J. Prentzas and I. Hatzilygeroudis. *Categorizing approaches combining rule-based and case-based reasoning*. Expert Systems, 24(2):97–122, 2007.

[22] S. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach*. Prentice Hall, 2nd edition, 2003.

[23] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, and M. Marchetti. *CAS-Mine: providing personalized services in context-aware applications by means of generalized rules*. Knowledge and Information Systems, 28(2):283–310, 2011.

[24] M. Strobbe, O. V. Laere, B. Dhoedt, F. D. Turck, and P. Demeester. *Hybrid reasoning technique for improving context-aware applications*. Knowledge and Information Systems, 31(3):581–616, 2012.

[25] J. Hong, E.-H. Suh, J. Kim, and S. Kim. *Context-aware system for proactive personalized service based on context history*. Expert Systems with Applications, 36(4):7448–7457, 2009.

[26] N. Bricon-Souf and C. R. Newman. *Context awareness in health care: A review*. International Journal of Medical Informatics, 76(1):2–12, 2007.

[27] U. Varshney. *Chapter 11: Context-awareness in Healthcare*. In Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring, pages 231–257. Springer Science + Business Media, LLC;, New York, NY, USA, 1st edition, 2009.

[28] J. Bardram. *Applications of context-aware computing in hospital work - Examples and design principles*. In Proceedings of the Annual ACM Symposium on Applied Computing, pages 1574–1579, Nicosia, Cyprus, 14-17 March 2004. New York, NY, USA: ACM Press;.

[29] M. B. Skov and R. T. Hoegh. *Supporting information access in a hospital ward by a context-aware mobile electronic patient record*. Personal and Ubiquitous Computing, 10(4):205–214, 2006.

[30] S. Mitchell, M. D. Spiteri, J. Bates, and G. Coulouris. *Context-aware multimedia computing in the intelligent hospital*. In Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system, pages 13–18, Kolding, Denmark, 17-20 September 2000. New York, NY, USA: ACM;.

[31] V. Stanford. *Beam me up, doctor McCoy*. IEEE Pervasive Computing, 2(3):13–18, 2003.

[32] M. A. Munoz, M. Rodríguez, J. Favela, A. I. Martinez-Garcia, and V. M. González. *Context-Aware Mobile Communication in Hospitals*. Computer, 36(9):38–46, 2003.

[33] K. Fishkin, M. Wang, K. P. Fishkin, and M. Wang. *A flexible, low-overhead ubiquitous system for medication monitoring*. Technical report, Intel Research Seattle, Technical Memo IRS-TR-03-011, 2003.

[34] C. Floerkemeier and F. Siegemund. *Improving the Effectiveness of Medical Treatment with Pervasive Computing Technologies*. In Proceedings of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at International Conference on Ubiquitous Intelligence and Computing, Seattle, Washington, USA, 25 October 2003.

[35] I. Korhonen, P. Paavilainen, and A. Särelä. *Application of ubiquitous computing technologies for support of independent living of the elderly in real life settings*. In Proceedings of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at International Conference on Ubiquitous Intelligence and Computing, Seattle, Washington, USA, 25 October 2003.

[36] P. de Toledo, S. Jimenez, F. del Pozo, J. Roca, A. Alonso, and C. Hernandez. *Telemedicine Experience for Chronic Care in COPD*. IEEE Transactions on Information Technology in Biomedicine, 10(3):567–573, 2006.

[37] B. Hu, B. Hu, J. Wan, M. Dennis, H.-H. Chen, L. Li, and Q. Zhou. *Ontology-based ubiquitous monitoring and treatment against depression*. Wireless Communications & Mobile Computing, 10(10):1303–1319, 2010.

[38] A. Mihailidis, B. Carmichael, J. Boger, and G. Fernie. *An intelligent environment to support aging-in-place, safety, and independence of older adults with dementia*. In Proceedings of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at International Conference on Ubiquitous Intelligence and Computing, Seattle, Washington, USA, 25 October 2003.

[39] T. Suzuki and M. Doi. *LifeMinder: an evidence-based wearable healthcare assistant*. In M. Beaudouin-Lafon and R. J. K. Jacob, editors, Proceedings of the ACM Conference on Human Factors in Computing Systems, pages 127–128, Seattle, Washington, USA, 31 March - 5 April 2001. New York, NY, USA: ACM.

[40] B. Jansen and R. Deklerck. *Context aware inactivity recognition for visual fall detection*. In Proceedings of the Pervasive Health Conference and Workshops, pages 1–4, Innsbruck, Austria, November 29 2006.

[41] V. F. S. Fook, S. C. Tay, M. Jayachandran, J. Biswas, and D. Zhang. *An ontology-based context model in monitoring and handling agitation behaviour for persons with dementia*. In Proceedings of the 4th IEEE Inter-

national Conference on Pervasive Computing and Commununications Workshops (PERCOMW), pages 560–564, Pisa, Italy, 13-17 March 2006. Washington, DC, USA: IEEE Computer Society;.

[42] D. Zhang, Z. Yu, and C.-Y. Chin. *Context-aware infrastructure for personalized healthcare*. Studies in Health Technology and Informatics, 117:154–163, 2005.

[43] F. Paganelli and D. Giuli. *An ontology-based system for context-aware and configurable services to support home-based continuous care*. IEEE Transactions on Information Technology in Biomedicine, 15(2):324–333, 2011.

[44] F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck. *An ontology-based nurse call management system (oNCS) with probabilistic priority assessment*. BMC Health Services Research, 11(26):28, 2011.

[45] T. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199–220, 1993.

[46] D. L. McGuinness and F. v. Harmelen. *OWL Web Ontology Language Overview*. Technical Report REC-owl-features-20040210, World Wide Web Consortium, February 10 2004. http://www.w3.org/TR/owl-features/.

[47] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[48] A. Valls, K. Gibert, D. Sánchez, , and M. Bateta. *Using ontologies for structuring organizational knowledge in Home Care assistance*. Internation Journal of Medical Informatics, 79(5):370–387, 2010.

[49] F. Ongenae, F. De Backere, K. Steurbaut, K. Colpaert, W. Kerckhove, J. Decruyenaere, and F. De Turck. *Appendix B: overview of the existing medical and natural language ontologies which can be used to support the translation process*. BMC Medical Informatics and Decision Making, 10(3):4, 2011.

[50] A. L. Rector, J. E. Rogers, P. E. Zanstra, and E. van der Haring. *OpenGALEN: Open Source Medical Terminology and Tools*. In Proceedings of the annual American Medical Informatics Association (AMIA) Symposium, page 982, Washington, DC, USA, 8-12 November 2003. American Medical Informatics Association (AMIA);. http://www.opengalen.org/.

[51] C. Rosse and J. L. V. M. Jr. *The Foundational Model of Anatomy Ontology*. In A. Burger, D. Davidson, and R. Baldock, editors, Anatomy Ontologies for Bioinformatics: Principles and Practice, pages 59–117. Springer;, London, UK, 2008.

[52] J. A. Blake and M. A. Harris. *The Gene Ontology (GO) project: structured vocabularies for molecular biology and their application to genome and expression analysis*. Current Protocols in Bioinformatics, 23(7.2.1-7.2.9):1472–6947, 2008. http://www.geneontology.org/.

[53] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, A. Jacobs, M. Van Gils, A. Ackaert, S. De Zutter, P. Verhoeve, and F. De Turck. *Participatory Design of a Continuous Care Ontology: Towards a User-Driven Ontology Engineering Methodology*. In J. Filipe and J. L. G. Dietz, editors, Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, 26-29 October 2011. ScitePress Digital Library;.

[54] M. D. Rodríguez, M. Tentori, J. Favela, D. Saldaña, and J.-P. García. *CARe: An Ontology for Representing Context of Activity-Aware Healthcare Environments*. In Proceedings of the AAAI Workshop on Activity Context Representation: Techniques and Languages, Paris, San Francisco, CA, USA, 7-8 August 2011. Menlo Park, USA: AAAI Press;.

[55] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 2nd edition, 2003.

[56] I. H. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan-Kaufmann, 3rd edition, 2011.

[57] M. Strobbe, J. Hollez, G. D. Jans, O. V. Laere, J. Nelis, F. D. Turck, B. Dhoedt, P. Demeester, N. Janssens, and T. Pollet. *Design of CASP: an open enabling platform for context aware office and city services*. In T. Pfeifer, J. Strassner, and S. Dobson, editors, Proceedings of the 4th International Workshop on Managing Ubiquitous Communications and Services (MUCS 2007), pages 123–142, Munich, Germany, May 25 2007.

[58] M. Strobbe, O. V. Laere, F. Ongenae, S. Dauwe, B. Dhoedt, F. D. Turck, P. Demeester, and K. Luyten. *Novel applications integrate location and context information*. IEEE PERVASIVE COMPUTING, 11(2):64–73, 2012.

[59] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL reasoner*. Journal of Web Semantics, 5(2):51–53, 2007.

[60] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. *Jena: implementing the semantic web recommendations*. In Proceedings of the 13th international conference on World Wide Web, Alternate track papers & posters, pages 74–83, New York, NY, USA, May 17–22 2004.

[61] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Technical report, 2004. http://www.w3.org/Submission/SWRL/.

[62] E. Prud'hommeaux and A. Seaborne. *SPARQL Query Language for RDF*. W3C Recommendation REC-rdf-sparql-query-20080115, January 15 2008. http://www.w3.org/TR/rdf-sparql-query/.

[63] P. Klinov. *Pronto: A Non-monotonic Probabilistic Description Logic Reasoner*. In Proceedings of the 5th European Semantic Web Conference, pages 822–826, Tenerife, Spain, June 1-5 2008.

[64] M. G. Davies and P. O. Hagen. *Systematic inflammatory response syndrome*. The British Journal of Surgery, 84(7):920–935, 1997.

[65] P. O. Nyström. *The systemic inflammatory response syndrome: definitions and aetiology*. Journal of Antimicrobial Chemotherapy, 41:1–7, 1998.

[66] M. J. O'Connor and A. K. Das. *A lightweight model for representing and reasoning with temporal information in biomedical ontologies*. In International Conference on Health Informatics (HEALTHINF), pages 90–97, Valencia, Spain, 2010.

[67] S. Verstichel, E. De Poorter, T. De Pauw, P. Becue, B. Volckaert, F. De Turck, I. Moerman, and P. Demeester. *Distributed ontology-based monitoring on the IBBT WiLab.t infrastructure*. In Proceedings of the 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), pages 509–525, Berlin, Germany, 2010.

[68] F. Ongenae, F. De Backere, K. Steurbaut, K. Colpaert, W. Kerckhove, J. Decruyenaere, and F. De Turck. *Appendix B: overview of the existing medical and natural language ontologies which can be used to support the translation process*. BMC Medical Informatics and Decision Making, 10(3):4, 2011.

[69] S. B. Kotsiantis. *Supervised Machine Learning: A Review of Classification Techniques*. Informatica, 31(3):249–268, 2007.

[70] S. B. Kotsiantis. *Decision trees: a recent overview*. Artificial Intelligence Review, 39(4):261–283, 2013.

[71] J. R. Quinlan. *Induction of decision trees*. Machine Learning, 1(1):81–106, 1986.

[72] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, USA, 1993.

[73] J. R. Quinlan. *Improved use of continuous attributes in C4.5*. Journal of Artificial Intelligence Research, 4(1):77–90, 1996.

[74] B. S. Everitt and A. Skrondal. *The Cambridge Dictionary of Statistics*. Cambridge University Press, New Yor, USA, 4th edition, 2010.

[75] T. Lukasiewicz. *Probabilistic Description Logics for the Semantic Web*. Technical report, Technical University of Wien, Institute for Information Systems, Wien, Austria, 2007.

[76] J. Su and H. Zhang. *A Fast Decision Tree Learning Algorithm*. In Proceedings of the 21st National Conference on Artificial Intelligence, pages 500–505, Boston, MA, USA, 2006.

6

A Self-learning Nurse Call System

F. Ongenae, M. Claeys, W. Kerckhove, T. Dupont, P. Verhoeve, and F. De Turck.

Submitted to Computers in Biology and Medicine, July 2013.

"We demand rigidly defined areas of doubt and uncertainty!"

– Douglas Adams, The Hitchhiker's Guide to the Galaxy (1952 - 2001)

This chapter further demonstrates the applicability of the self-learning framework presented in the previous chapter by applying it to the oNCS, which was discussed in Chapter 4. The self-learning framework is used to automatically adapt the parameters of the probabilistic priority algorithm, i.e., the defined probabilities and thresholds, to the needs and requirements of the specific department where the oNCS is deployed. Thus, while Chapter 5 focusses on how the self-learning framework can be used to discover completely new knowledge, this chapter concentrates on how it can be employed to adapt existing knowledge. This research is thus related to Research Contributions 4 and 5 discussed in Section 1.3 of Chapter 1.

⋆ ⋆ ⋆

Abstract In recent years the complexity of care in institutionalized care settings has increased due to an ageing population in need of more complex care, a dwindling number of caregivers requiring a more efficient use of resources and increasing healthcare costs. Electronic healthcare (eHealth) solutions are often introduced

to deal with these issues. However, the introduction of all this technological equipment further increases the complexity of healthcare as the caregivers are responsible for integrating and configuring the eHealth solutions to suit their needs. Small differences in user requirements often occur between different environments where the eHealth services are deployed. It is difficult to capture these small nuances at development time as domain expert often find it difficult to assess these parameters. Consequently, the services are not tuned towards the needs of the users and they are required to change their behavior to accommodate the technology instead of the other way around.

This paper describes our experiences with extending an electronic healthcare application with self-learning components such that it can automatically adjust its parameters at run-time to the needs and preferences of the users. These self-learning components first gather information about the usage of the application. This collected information is processed by data mining techniques to learn the parameter values for the application. Each discovered parameter is then associated with a probability, which expresses its reliability. Unreliable values are filtered. The remaining parameters and their reliability are integrated into the application.

The used eHealth application is an ontology-based Nurse Call System (oNCS), which assesses the priority of a call based on the collected healthcare and context data and uses this information to assign the most appropriate caregiver to a call. The self-learning components use decision trees and Bayesian networks to automatically learn and adjust the parameters of the priority algorithm of the oNCS. It is shown that for a realistic dataset of 1,050 instances, correct parameter values are discovered. These new parameters are also learned very efficiently as the components require at most 100 milliseconds (ms) execution time and 20 megabyte (MB) memory.

6.1 Introduction

Worldwide the proportion of people aged over 60 years is growing faster than any other age group, as a result of both longer life expectancy and declining fertility rates [1]. Because of health problems, a lot of the elderly are no longer able to live independently and require some form of institutionalized long-term care, e.g., residential care or long stays in the hospital [2]. These developments are accompanied by emerging staff shortages in the formal care sector. In 2006, the World Health Organization (WHO) reported an estimated shortage of almost 4.3 million doctors, midwives, nurses and support workers worldwide [3]. Moreover, people are increasingly living longer with one or more chronic diseases, which increases the complexity of diagnosis and treatment and requires more personalized healthcare and specialized staff. Consequently, the healthcare costs have also been on the rise. Spending on healthcare almost consistently grows faster than the Gross

Domestic Product (GDP) [4].

To achieve a more optimized use of resources and rostering of staff and to reduce the healthcare costs, Information Technology (IT) and technological equipment, e.g., monitoring equipment and Electronic Patient Records (EPR), are often introduced in institutionalized healthcare settings [5]. Electronic Healthcare (eHealth) software and services can then be built that take advantage of all the collected information to ideally support caregivers in their daily work practices. The benefits of eHealth, such as improved operational efficiency, higher quality of care, and positive return on investments have been well documented in the literature [6]. However, the increased introduction of eHealth also increases the complexity of healthcare as the caregivers are responsible for tweaking and configuring the eHealth solutions to suit their needs. The various healthcare environments where the services are deployed, e.g., different nursing units or hospital departments, have slightly different requirements pertaining to how the collected information about the patients, caregivers and environment is taken into account. It is difficult to capture these small nuances at development time as domain experts often find it difficult to assess these parameters. Consequently, the resulting services are not really personalized towards the needs and preferences of the caregivers and they have to significantly alter their workflow patterns to accommodate the technology instead of the other way around [7]. This hinders the adoption of these services [8].

An important way to coordinate work, communicate and provide continuous care is by making use of a nurse call system. In previous research, we have developed an ontology-based Nurse Call System (oNCS) [9], which finds the most appropriate caregiver to handle a call based on profile and environment information captured in an ontology, e.g., the risk factors of the patient, the locations of the staff and patient, the priority of the call and the current tasks of the staff. Simulations showed that the workload distribution amongst nurses and the arrival times of caregivers at calls are positively influenced by using the oNCS [9]. However, user tests performed with the prototype also showed that small nuances were often required in how the profile information was taken into account within a specific healthcare setting. Domain experts also found it difficult to specify the parameters of the oNCS, i.e., which context should be taken into account and how, at development time. To resolve this issue, this paper presents an extension of the oNCS that allows automatically adjusting its parameters at run-time.

The remainder of this paper is structured as follows. Section 6.2 gives an overview of the oNCS and the associated priority assessment and nurse call algorithm. Section 6.3 details the extension of the oNCS with components, which enable the autonomous adjustment of its parameters. The implementation of these components is discussed in Section 6.4, while Section 6.5 highlights how the correctness and performance of the extension was evaluated. Finally, Section 6.6

discusses the results and Section 6.7 summarizes the conclusions.

6.2 Ontology-based Nurse Call System

The main functionality of the oNCS is to provide efficient support for wireless nurse call buttons and to employ a sophisticated nurse call algorithm that takes the profiles of the staff members and patients into account. A detailed description can be found in Ongenae, et al. [9]. To realize the latter, a continuous care ontology [10] is used of which the most important classes pertaining to the dynamic algorithm are visualized in Figure 6.1. An ontology [11] formally models all the concepts and their relationships and properties within a domain. The ontology models people and associates them with their roles, location, profile, the hospital department they work or lie on, risk factors, and current tasks. Additionally, the ontology models the various types of nurse calls. Patients can launch three types of calls, i.e., service calls for "caring" requests, sanitary calls originating from sanitary spaces and normal calls for mostly medical requests. All the other calls, i.e., urgency, medical, technical and (sanitary) assistance calls, are launched by nurses. Each call is associated with a status and a priority. It is also indicated who made the call and which staff members are assigned to it.

When a new call is launched, the information captured in the ontology is used to assign the most appropriate staff member to the call. First, the priority of the call is determined, using the algorithm visualized in Figure 6.2. The ontology specifies for each risk factor a probability, which indicates the likelihood that a patient with this risk factor is classified as a high, medium or low risk patient. Patients can of course exhibit several risk factors. In this case, probabilistic reasoning on the speficied probabilities is used to determine for each risk group the combined likelihood that a particular patient belongs to it. As shown in Figure 6.1, there are seven priority levels. Probabilities are indicated in the ontology, which specify the likelihood that a call of a particular type made for a patient associated with a particular risk group has a certain priority. As example, Table 6.1 shows the probabilities for the types of calls, which can be launched by patients. For each of the seven priority classes, probabilistic reasoning is thus used to combine these probabilities with the probabilistic assignment of patient to risk groups in order to determine the likelihood that a call of a certain type has this priority. To determine the suitable priority for this call based one these probabilistic values, a threshold algorithm is used. Thresholds are specified in the ontology for each priority class. If the probabilistic value for the highest priority is higher than or equal to the threshold for this priority, the call is associated with the highest priority. If not, the same condition is checked for the other priority classes in the following order: high, above normal, below normal, normal, low and lowest.

The priority of the call is then combined with the other context information in

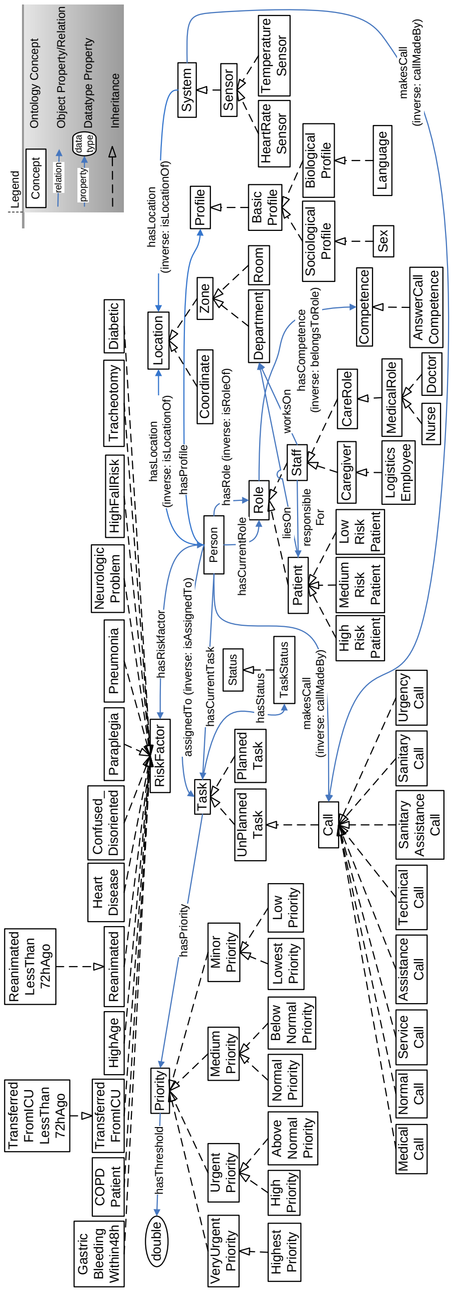


Figure 6.1: Prevalent concepts of the continuous care ontology used by the oNCS.

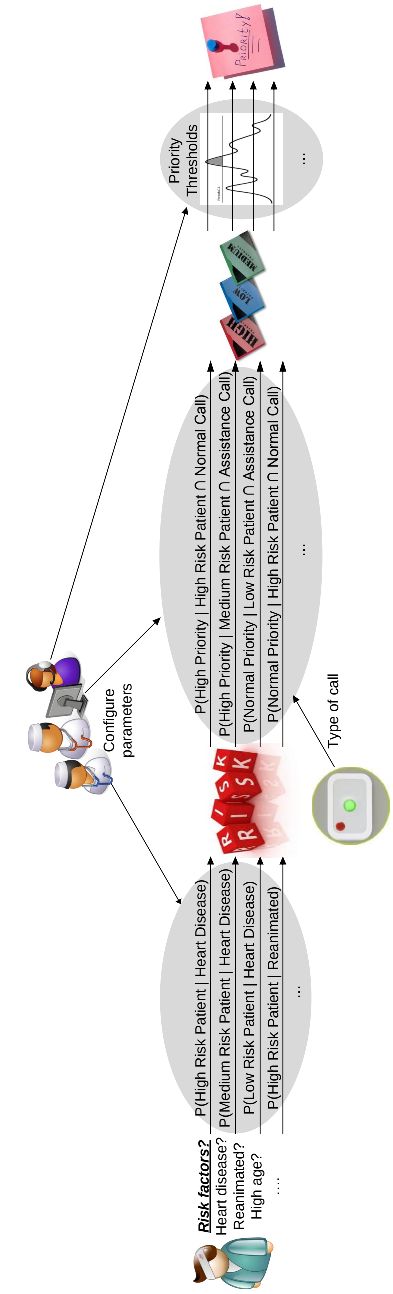


Figure 6.2: Probabilistic priority algorithm

Risk group	Type of call	Highest	High	Above normal	Normal	Below Normal	Low	Lowest
High	Normal		0.2	0.6	0.2			
	Sanitary		0.3	0.6	0.1			
	Service			0.2	0.2	0.6		
Medium	Normal			0.3	0.6	0.1		
	Sanitary			0.4	0.5	0.1		
	Service				0.2	0.4	0.4	
Low	Normal				0.6	0.3	0.1	
	Sanitary				0.7	0.2	0.1	
	Service					0.4	0.4	0.2

Table 6.1: Probabilistic assignment of priorities to calls based on the risk group of the patient and the type of call.

the ontology to find the most appropriate staff member to handle the call, e.g., the distance between the caregivers and the patient, the current tasks of the available staff and the capability of the caregivers to handle the call based on their roles and competences. For calls with a higher priority, more weight is given to finding a caregiver who is able to quickly rush to the patient and assess the situation. In contrast, other context information is given more weight for calls with a lower priority such as the profile and competences of the staff. The assigned caregiver receives the call on a smartphone, which runs the mobile nurse call application. This application allows staff to receive, assess, accept and redirect calls. They are also able to change the priority of the call or indicate its reason. The information provided by the caregivers using the application is also captured in the ontology.

It can be noted that the adequate assessment of the priority of a call and thus the suitable assignment of caregivers to calls, largely depends on the correctness of the specified probabilities and thresholds. The probabilities were determined by consulting various domain experts, i.e., nurses, doctors and developers of nurse call systems. The thresholds were determined by running simulations of calls and calculating the probabilistic priority assignment for these calls using the probabilities defined by the experts. Thresholds were then chosen such that the distribution of the simulated calls across the different priority classes deviates the least from the ideal distributions as determined by the experts, namely 5% - 10% - 25% - 35% - 25% - 0% - 0%, ordered from the highest to the lowest priority.

However, it was found that domain experts struggled with defining these probabilities and ideal distribution of calls amongst priority categories. It was also difficult to extract these probabilities out of logging data as the current installed nurse call systems do not allow nurses to indicate or change the priority of a call. Furthermore, these parameters also slightly differ between hospital departments depending on the medical profile of the patients and the gravity of the treated pathologies. Therefore it was chosen to initialize the oNCS with the educated guesses of the domain experts and employ a self-learning framework. This framework allows automatically adjusting the probabilities and thresholds to the specific needs of the department where the oNCS is deployed.

6.3 Self-learning extension of the oNCS

The self-learning extension of the oNCS is visualized in Figure 6.3. The oNCS was built as an extension of the *Context-Aware Service Platform (CASP)* [12], which consists of a collection of *OSGi* [13] bundles to handle context information. The *Context Framework Layer* contains the *Context Interpreter*, which uses the continuous care ontology implemented in OWL [14] to model all the context information gathered about the environment, tasks, calls, patients and staff members. Pronto [15] is used to reason on the probabilistic information in the ontology, while Jena Rules [16] implement the threshold and nurse call algorithm. The *Context Providers* allow inserting new information into the *Knowledge Base*, e.g., a new nurse call or location of the patient. This new information can come from a database (*Persistence Layer*) or directly from a device (*Device Layer* and *Context Gathering Layer*). In contrast, the *Query Services* are used to extract derived knowledge from the *Knowledge Base*, such that it can be processed by the applications and services in the *Application Layer*. To improve the scalability and robustness of the system, context information can be stored in the *Persistence Layer*. This historical context information can then be exploited by the new self-learning components to adjust the parameters of the oNCS to the behavior of the users. These new components are indicated in grey.

The *Monitoring Component* constantly monitors the ontology to pick up trends and patterns in the way the priorities are assigned to calls by the caregivers. This component stores the evidence in the *Persistence Layer*. This evidence can be inspected by the domain experts by using the *Configuration Module*. When enough evidence has been collected, the *Learning Pipeline* can be initiated by the *Configuration Module*. The *Configuration Module* is notified of which data should be collected for the *Learning Pipeline*, either by the *Monitoring Component* or by the domain experts and administrator. The latter allows to initiate the *Learning Pipeline* with external data provided by the stakeholders. The *Configuration Module* configures the Pipeline Manager to use the *Data Collection Component*, *Input Convertor* and *Integration Component* that suits this type of evidence. It also passes the correct parameters to the *Pipeline Manager*, which are needed to

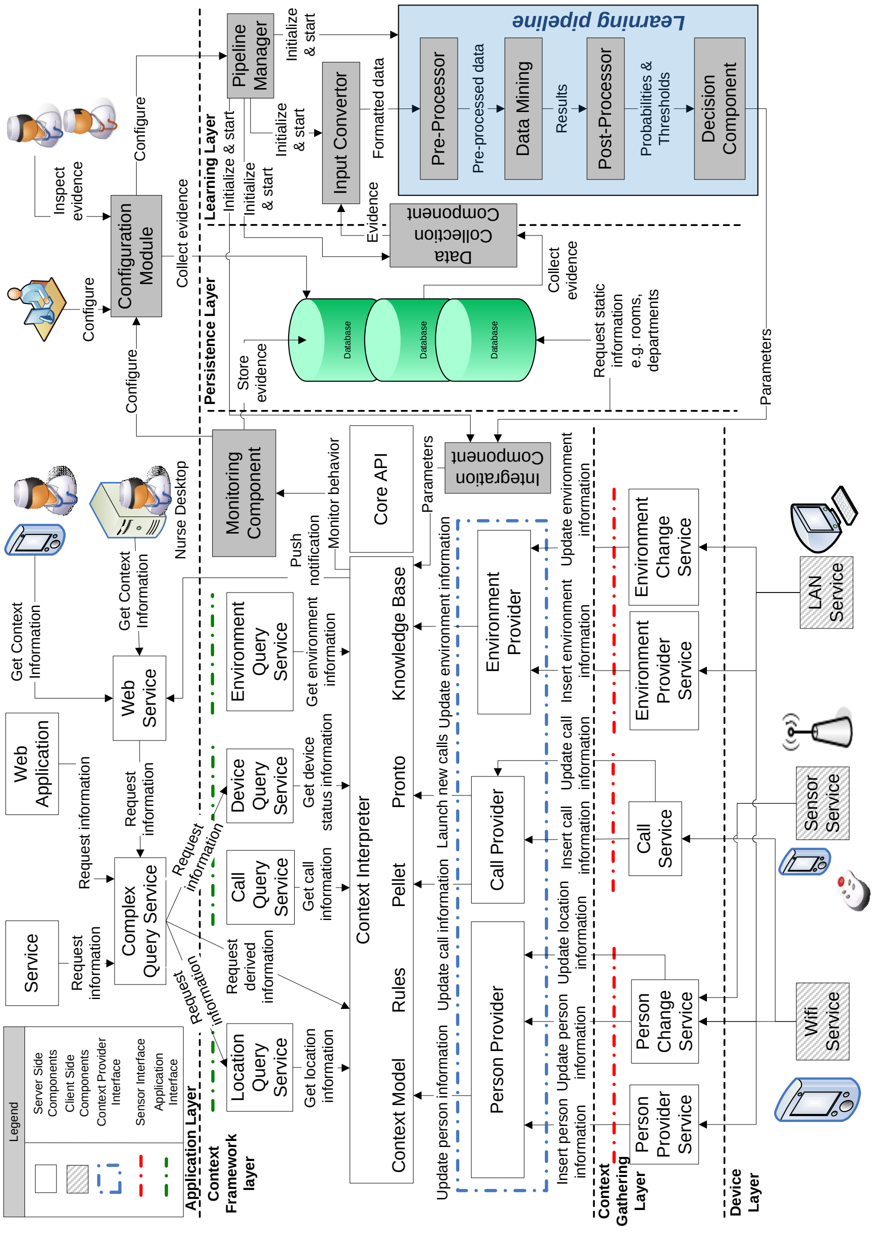


Figure 6.3: The oNCS extended with self-learning components

retrieve the data from the *Persistency Layer* using the *Data Collection Component*.

The *Learning Pipeline* is implemented using the Pipes-and-Filters architectural design pattern [17]. A pipeline consists of a set of filters, implementing small processing steps, which are connected by pipes. All the filters implement the same interface such that they can easily be rearranged, omitted or added. In this way, an extensible and flexible architecture is achieved.

The *Pipeline Manager* initiates the *Data Collection Component* to collect the necessary evidence. To achieve a flexible *Learning Pipeline*, a generic internal data format is used, which allows expressing both the information which is used as evidence and the probabilities and thresholds that are obtained as output. The format is largely based on the Attribute-Relation File Format (ARFF), which is the text file format used by the Waikato Environment for Knowledge Analysis (WEKA) [18]. The *Input Convertor* is responsible for converting the collected data to this format.

Next, the *Pipeline Manager* creates and starts the *Learning Pipeline*. *Pre-Processor* components can be used to clean the data, e.g., remove outliers or scale the data. This cleaned data is then processed by a *Data Mining* component to build a model, e.g, a Bayesian network or decision tree, that conveys the relation between the properties of the call, e.g., its type and the patient group, and it priority. This learned model is then processed by a *Post-Processor* component to extract the probabilities or thresholds for the oNCS.

Finally, to assess the correctness of the learned probabilities and thresholds, the *Decision Component* associates each discovered parameter with a probabilistic value expressing its reliability. When the calculated probabilistic value is too low, the discovered parameter is discarded and not adjusted in the oNCS.

The *Integration Component* is responsible for adjusting the parameters of the oNCS according to the probabilities and thresholds discovered by the *Learning Pipeline*. The associated probability, which was calulated by the *Decision Component*, is also added to the ontology to convey the reliability of the parameter values to the domain experts. If the parameter value in the ontology is the same as the learned value, the associated probability is updated to reflect its increased reliability, namely by using the average of the old and new probability.

6.4 Implementation details

Two scenarios can be identified, namely adjusting the probabilities and the thresholds. For the first scenario, this paper focuses on adjusting the probabilities, which indicate that a call has a particular priority based on its type and the risk group of the patient, who made the call. We will concentrate on learning the probabilities for calls launched by patients, i.e., normal, service and sanitary calls. Ajdusting the probabilities that indicate the likelihood that patients belong to particular risk groups and for other types of calls, is analogous. The pipelines for these scenarios are visualized in Figures 6.4 and 6.5.

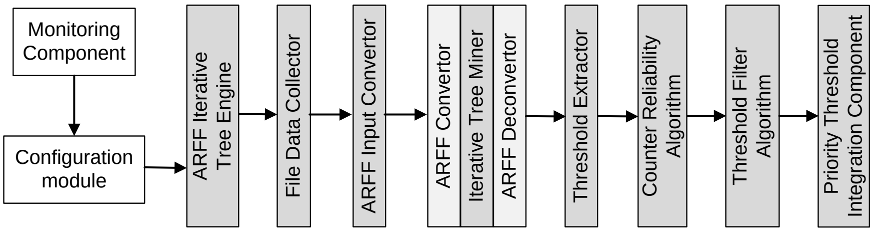


Figure 6.4: The Learning Pipeline used to learn and adjust the threshold parameters of the oNCS

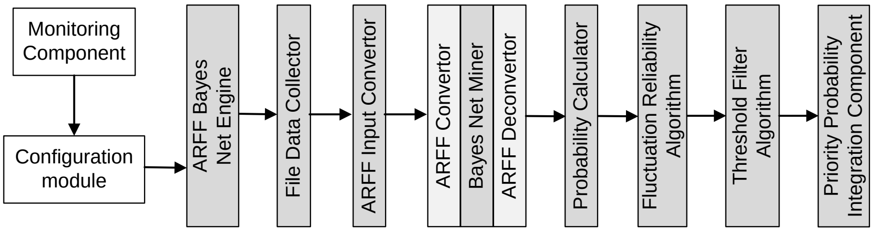


Figure 6.5: The Learning Pipeline used to learn and adjust the probabilistic parameters of the oNCS

Higest priority	High priority	Above Normal priority	Normal priority	Below Normal priority	Low priority	Lowest priority	Assigned priority
0.13	0.29	0.25	0.07	0.03	0.81	0.27	Above normal
0.18	0.96	0.46	0.45	0.06	0.66	0.01	High
0.12	0.18	0.20	0.00	0.00	0.00	0.70	*Below normal*
0.07	0.05	0.88	0.27	0.18	0.12	0.12	Above normal
0.06	0.02	0.15	0.11	0.02	0.56	0.59	Normal
0.44	0.11	0.53	0.27	0.21	0.51	0.31	Highest
0.20	0.09	0.12	0.01	0.04	0.54	0.03	*Above normal*

Table 6.2: Some example instances of the dataset to learn the threshold parameters

6.4.1 Data collection and input conversion

The *Monitoring Component* monitors the ontology for new calls that receive the status `Finished`, indicating that the call has been completely handled and processed by the caregiver. The component collects the type and priority of the call using SPARQL [19] queries. The priority can be the one assigned by the oNCS, but it is also possible that the caregiver changed it using the mobile nurse call application. The *Monitoring Component* also retrieves the probabilistic assignment of the call to the seven priority classes based on its type and the probabilistic assignment of the patient to the three risk groups using the probabilistic reasoner Pronto. Finally, the probabilistic assignment of this patient to the three risk groups

Risk group	Type of call	Assigned priority
High	Normal	Above normal
Low	Sanitary	Low
Medium	Normal	Normal
High	Service	*High*

Table 6.3: Some example instances of the dataset to learn the probability parameters of the assignment of calls to priority classes

is requested. Based on this collected data, two datasets are created. Each instance in the dataset represents one call. The first is used to learn the threshold parameters and contains for each call the calculated probabilistic value for each priority class and the priority that was assigned it. Some example instances of this dataset are illustrated in Table 6.2. The second dataset is used to learn the probabilistic assignment of calls to priority classes based on their type and the risk group of the patient associated with the call. It indicates for each call the risk group of the patient, the type of the call and the assigned priority. Only calls with type normal, service or sanitary are retained. The risk group for the patient is chosen based on the calculated probabilistic assignment of this patient to the risk groups. For example, a patient with a heart disease has at least 50% chance of being a high risk patient. Some example instances of this dataset are listed in Table 6.3. To be able to demonstrate the *Input Convertor*, the datasets are saved in the ARFF format in the *Persistence Layer*.

The *Monitoring Component* keeps track of how many instances have been collected for each dataset. When a representative amount has been gathered, the *Configuration Module* is invoked to initiate the *Learning Engine*. Different *Learning Pipelines* are used to process each of the scenarios. These are implemented by different *Pipeline Managers*, e.g., *ARFFBayesNetEngine* or *ARFFIterativeTreeEngine*. The *Monitoring Component* also indicates to the *Configuration Module* the location of the data, its format and which *Pipeline Manager* should be used.

The *Configuration Module* configures the *Pipeline Manager* to use the appropriate *Data Collection Component* and *Input Convertor*, wich suit the format of the data. A *File Data Collector* was implemented, which is able to read the data from a file at a specified location. The result is a `String`, which is provided to the *ARFF Input Convertor*. This *Input Convertor* is able to translate this `ARFF-String` to the internal format used by the *Learning Pipeline*. A *Pre-Processor* is not needed for these scenarios as no anomalies can occur in the data.

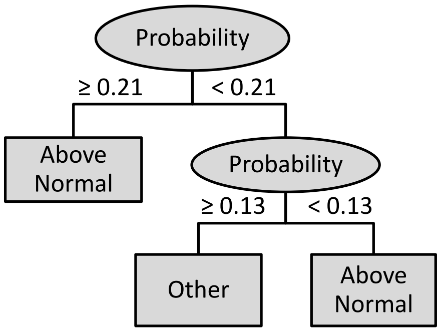


Figure 6.6: Example of a decision tree that encodes the learned knowledge about the threshold for the Normal priority class

6.4.2 Data mining & post-processing

Both scenarios use the WEKA data mining toolbox to learn the thresholds and probabilities of the oNCS. The first uses decision trees [20], while the latter uses a Bayesian network [21]. The following subsections detail how these models are built and how the parameters of the oNCS are derived from them. As previously mentioned, WEKA uses the ARFF data format to represent data. Therefore, (de)convertors were implemented that are able to translate the internal data format of the *Learning Pipeline* to and from the ARFF data format.

6.4.2.1 Discovering the thresholds using a C4.5 decision tree

The *Data Mining* filter needs to find relations in the threshold dataset between the probabilistic assignment of the calls to the priority classes and the priority that was eventually assigned to the calls. The former are considered input attributes, while the latter is called the label. Supervised [18] classification techniques [22] are used to discover these relations between the input attributes and the label. Decision trees are a well-known and easy to use classification technique. A decision tree consists of leaves, which each represent a possible value of the label, and internal nodes and branches, which represent the attributes on which the decision is based and the conditions that they must fulfill. An example is visualized in Figure 6.6. For this research, the J4.8 Java implementation of the C4.5 algorithm [23] in the WEKA data mining tool was used to build the decision trees.

The following knowledge about the threshold algorithm can be exploited to optimize the data mining. First, a call is assigned a priority x based on the probabilistic value for this priority class. Second, the probabilistic values for the priority classes are checked in a particular order, as discussed in Section 6.2. The probabilistic values for the priority classes, which occur later in the sequence than the assigned priority, are not taken into account for this call. Consequently, the deci-

Above Normal priority	Assigned priority
0.25	Above normal
0.20	Other
0.88	Above normal
0.15	Other
0.12	Above normal

Table 6.4: Some example instances of the dataset to learn the threshold parameter for the Normal priority class

sion was made to implement an *Iterative Decision Tree* algorithm, which builds a separate decision tree for each priority class. The decision trees are built in the same order as the priority classes are checked by the threshold algorithm. The used dataset for each iteration consists only of one input attribute, i.e., the priority class under scrutiny. The label can also only assume two values, namely the considered priority and "Other". The latter is used to replace all other possible priority classes. Finally, all the instances that were assigned a priority class, which is checked earlier than the priority class for which the decision tree is being built, are removed from the dataset. In this way, a dataset is built, which can be used by a decision tree to learn when the probabilistic value of a priority class is high enough to receive this priority as label. As an example, Table 6.4 visualizes some instances of the dataset for the Above Normal priority class, which were derived from the original dataset visualized in Table 6.2. It can be noted, that all the instances were removed, which were assigned the Highest and High priority, as these are checked earlier by the threshold algorithm.

The *Iterative Decision Tree* algorithm builds the decision tree for each priority class. The J4.8 algorithm outputs a textual representation of the decision tree. For example, the tree visualized in Figure 6.6 is represented as follows:

N0 [label="Probability"]

N0 $\rightarrow$ N1 [label=" $>=$ 0.21"]

N1 [label="Above Normal"]

N0 $\rightarrow$ N2 [label=" $<$ 0.21"]

N2 [label="Probability"]

N2 $\rightarrow$ N3 [label=" $>=$ 0.13"]

N3 [label="Other"]

N3 $\rightarrow$ N4 [label=" $<$ 0.13"]

N4 [label="Above Normal"]

The nodes and branches are identified and translated to the internal data format such that the results can be forwarded to the *Post-Processor*.

The *Threshold Extractor Post-Processor* was implemented, which extracts the discovered thresholds out of the textual representation of each decision tree. For each decision tree, all the branches are considered that result in a leaf with the priority class label, associated with this decision tree. The branches, which result in a leaf with the label “Other”, are ignored. All the considered branches are followed from the leaf up to the root and the conditions are checked. The condition that represents the highest lower bound is chosen as threshold for this priority class, i.e., a condition of the type $\geq x$ where x is the highest value for a condition of this type in this tree. The discovered thresholds are represented in the internal data format and forwarded to the *Decision Component*.

6.4.2.2 Discovering the probabilities using a Bayesian network

In this scenario, the *Data Mining* filter needs to find probabilistic relations between two input attributes, i.e., the type of the calls and the risk group of the patients, and the priority labels that were eventually assigned to the calls. Bayesian networks can ideally be used to discover these probabilistic relations. Bayesian networks are graphical models that represent the conditional dependencies between a set of variables as a directed acyclic graph. Each node is associated with a probability function. This function is able to calculate the probability of the variable respresented by this node based on a particular set of values for the variables, which are represented by nodes that are parents of this node. Different techniques can be used to build Bayesian networks. Naive Bayesian networks assume that all the input attributes are conditionally independent. Consequently, a network is obtained in which the label is connected to each input attribute, but the input attributes are not connected to each other. As the risk group of the patient is independent of the types of calls this patient makes, Naive Bayesian networks are used for this research. The BayesNet implementation of WEKA was used to construct the network. The probabilities obtained by building the network are retrieved from WEKA and represented in the internal data format.

The *Probability Calculator Post-Processor* was implemented to calculate the needed probability parameters for the oNCS. To explain this calculation, the following notation is introduced:

- The risk group input attribute is represented by A and has $n1$ possible values $a_1, ..., a_{n1}$.
- The type of call input attribute is depicted by B and has $n2$ possible values $b_1, ..., b_{n2}$.
- X represents the label, i.e., the priority class, and has m possible values $x_1, ..., x_m$.

The output of the BayesNet algorithm contains the following probabilities:

- $P(X = x_i), \forall\, i \in [1, m]$.
- $P(A = a_i|X = x_j), \forall\, i \in [1, n1]$ and $\forall\, j \in [1, m]$.
- $P(B = b_i|X = x_j), \forall\, i \in [1, n2]$ and $\forall\, j \in [1, m]$.

Bayes' rule can be used to calculate the probability parameters for the oNCS:

$$P(X = x_i|A = a_j \cap B = b_k) = \frac{P(A = a_j \cap B = b_k|X = x_i)P(X = x_i)}{P(A = a_j \cap B = b_k)}$$
$$\text{where } i \in [1, m], j \in [1, n1] \text{ and } k \in [1, n2] \quad (6.1)$$

Only the probabilities $P(X = x_i)$ can be directly derived from the Bayesian network. As attributes A and B are conditionally independent, the other term of the numerator can be calculated as follows:

$$P(A = a_j \cap B = b_k|X = x_i) = P(A = a_j|X = x_i)P(B = b_k|X = x_i)$$
$$\text{where } i \in [1, m], j \in [1, n1] \text{ and } k \in [1, n2] \quad (6.2)$$

The probabilities on the right hand side of this equation can also be derived from the Bayesian network. These calculated probabilities can be used to derive the denominator using the law of total probability as follows:

$$P(A = a_j \cap B = b_k) = \sum_{i=1}^{m} P(A = a_j \cap B = b_k|X = x_i)P(X = x_i)$$
$$\text{where } j \in [1, n1] \text{ and } k \in [1, n2] \quad (6.3)$$

By inputting the results of Equations 6.2 and 6.3 in Equation 6.1, the needed probability parameters can be calculated. These parameters are represented in the internal data format and forwarded to the *Decision Component*.

6.4.3 Filtering the results and expressing their reliability

As mentioned in Section 6.3, the *Decision Component* attaches probabilities to the discovered parameters to express their reliability to the users.

To assess the reliability of the thresholds, the *Counter Reliability Algorithm* is used. This algorithm applies the new thresholds to the original dataset. For all the calls of a particular priority, it then calculates the percentage that received this priority correctly by the new threshold algorithm. For example, suppose that 0.44 - 0.35 - 0.21 - 0.07 - 0.2 - 0 - 0 are discovered as thresholds, ordered from the Highest to the Lowest priority. If these thresholds are applied to the dataset visualized in Table 6.2, the threshold for the Above Normal priority achieves 67% reliability, as the first and fourth call are correctly assigned the Above Normal priority, while the last call incorrectly receives the Low priority.

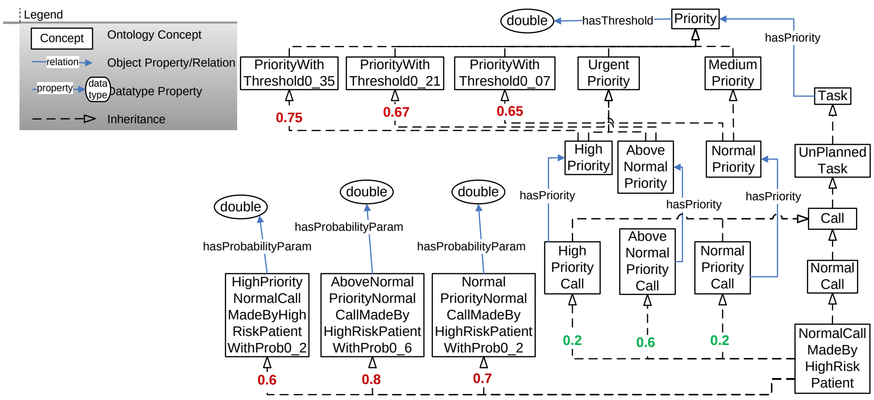


Figure 6.7: Integrating the learned parameters of the oNCS into the ontology with an associated probability to express their reliability

The *Fluctuation Reliability Algorithm* computes the reliability of the discovered probability parameters. It first calculates the difference x between the new and old parameter value. When the Learning Pipeline is used for the first time to learn the probability parameters, the probability parameters in the ontology are used as the old parameter values. In later runs of the pipeline, the parameter values discovered in the previous run are used as old parameter values. The reliability of the new parameter is then set to $1-x$. Consequently, if the *Learning Pipeline* consecutively discovers very similar parameter values, the reliability increases. The reliability thus increases if the value of the parameter converges.

A simple filter algorithm, namely the *Threshold Filter Algorithm*, was implemented, which filters the parameters for which the reliability is lower than a specified threshold, e.g., 50%. These parameters are not adjusted in the oNCS. However, these discovered parameters are stored such that they can be used by subsequent runs of the *Learning Pipeline*, e.g., as old parameter values in the *Fluctuation Reliability Algorithm*.

6.4.4 Integrating the parameters in the oNCS

6.4.4.1 Integrating the thresholds in the oNCS

The *Priority Threshold Integration Component* is responsible for integrating the discovered thresholds into the oNCS with their associated probability. To integrate a discovered threshold for a particular priority class, this component first checks whether this priority was already associated with this threshold, i.e., the parameter value has not changed. If this is the case, only the reliability is changed, as explained further. To integrate a new threshold, a subclass of the `Priority` class is introduced in the ontology, as shown in Figure 6.7. For example, to integrate the threshold of 0.21 for the Above Normal priority, the `PriorityWithThreshold0_21` class is created. This class is defined as follows:

Priority AND (hasThreshold VALUE 0.21^^double)

If this class already exists in the ontology, it is re-used. The priority class associated with this threshold is then defined as a subclass of this class, e.g., `AboveNormalPriority` becomes a subclass of `PriorityWithThreshold0_21`. The priority also inherits the definition and is thus effectively associated with the correct threshold. The subclass relationship to the previous threshold is removed.

Next, the associated reliability is expressed in the ontology. Pronto is used to represent and reason on the probabilistic information in the ontology. To express probabilistic knowledge, Pronto uses Generic Conditional Constraints (GCCs) [24]. A GCC is of the form (D—C)[l,u] where D and C are classes in the ontology and [l,u] is a closed subinterval of [0,1]. To represent these GCCs in the ontology, Pronto employs subsumption axiom annotations. For example, to express that the 0.21 threshold for the normal priority class only has a reliability of 67%, the subclass relationship between the `AboveNormalPriority` and `PriorityWithThreshold0_21` concepts is annotated as follows:

< owl11:Axiom >

< rdf:subject rdf:resource="#AboveNormalPriority" >

< rdf:predicate rdf:resource="&rdfs;subClassOf" >

< rdf:object rdf:resource="#PriorityWithThreshold0_21" >

< pronto:certainty > 0.67;0.67 < /pronto:certainty >

< owl11:Axiom >

Pronto uses probability intervals to express probabilistic knowledge. However, as illustrated in the previous example, strict probabilities can easily be expressed by defining an interval with an equal upper and lower limit. When a new threshold is associated with a priority, the reliability calculated by the *Decision Component* is used. If the priority was already connected to this threshold, the reliability is changed to the average of the old and the new reliability.

6.4.4.2 Integrating the probabilities in the oNCS

The probability parameters, which express the the likelihood that a call of a particular type made by a patient belonging to a specific risk group has a particular priority, are represented in the ontology by annotated subsumption axioms between `Call` classes, as illustrated in Figure 6.7. For example, the following annotated subsumption axiom expresses that a normal call made by a high risk patient has 0.2 probability of having a normal priority:

< owl11:Axiom >

< rdf:subject rdf:resource=“#NormalCallMadeByHighRiskPatient” >

< rdf:predicate rdf:resource=“&rdfs;subClassOf” >

< rdf:object rdf:resource=“#NormalPriorityCall” >

< pronto:certainty > 0.2;0.2 < /pronto:certainty >

< owl11:Axiom >

These two classes are defined as follows:

NormalCallMadeByHighRiskPatient:

NormalCall AND (callMadeBy SOME (hasRole SOME HighRiskPatient))

NormalPriorityCall:

Call AND (hasPriority SOME NormalPriority)

To integrate the discovered probability parameters in the oNCS, the *Priority Probability Integration Component* just changes the probabilistic value in the annotated subsumption axiom.

Next, the *Priority Probability Integration Component* associates the reliability with this discovered parameter. To realize this, a new class is created in the ontology that represents the annotated subsumption axiom. For example, to represent the previous subsumption axiom, the class `NormalPriorityNormal-CallMadeByHighRiskPatientWithProb0_2` was created with the following definition:

hasProbabilityParam VALUE 0.2˜double

An annotated subsumption axiom is then created, which associates the input attributes, i.e., a call of a particular type made by a patient belonging to a specific risk group, with this new class and annotates this subclass relationship with the reliability. For example, the following annotated subsumption axiom is created for the running example to express that this parameter value has a reliability of 70%:

< owl11:Axiom >

< rdf:subject rdf:resource=“#NormalCallMadeByHighRiskPatient” >

< rdf:predicate rdf:resource=“&rdfs;subClassOf” >

< rdf:object rdf:resource=“#NormalPriorityNormalCallMadeBy

HighRiskPatientWithProb0_2” >

< pronto:certainty > 0.7;0.7 < /pronto:certainty >

< owl11:Axiom >

Note that if the parameter value has not changed, the reliability is updated to 100%, as this reliability expresses how much the parameter value deviates from the previous value.

6.5 Evaluation set-up

To adequately evaluate the correctness and performance of the self-learning components, generated datasets are used for both scenarios. In this way, trends can be introduced into the datasets, which should be discovered by the *Learning Pipeline*. To achieve realistic datasets, noise is introduced. The following subsections detail how these datasets were generated and noise was added. The datasets were generated in the ARFF format and stored in the *Persistence Layer* so that they can be retrieved by the *File Data Collector* and translated to the internal format by the *ARFF Input Convertor*.

To evaluate the applicability of the framework, it is important to assess the correctness of the derived parameters. The correctness of the used data mining techniques is influenced by the size of the dataset and the amount of noise. To assess the influence of the latter, the *Learning Pipeline* was consecutively applied to datasets of the same size, but with an increasing amount of noise. The amount of noise is varied from 0% to 50% in steps of 1%. It is unnecessary to increase the noise percentage beyond 50% as a random label is assigned at this point and the dataset becomes meaningless. The amount of noise needs to be increased in a dataset of realistic size. Each instance in the dataset corresponds to one made by or for a patient. Out of logging data of the nurse call system installed at Ghent University Hospital [25], it was derived that one average five calls are made per 24 hours by or for a specific patient. Consequently, for a nursing unit containing on average 30 patients, 1,050 calls are launched per week on average. Therefore, to assess the influence of noise, datasets were generated containing 1,050 instances.

The influence of the size of the dataset on the correctness is evaluated by consecutively applying the *Learning Pipeline* to datasets of increasing size. The dataset sizes range from 100 to 2,000 instances in steps of 100 instances. This range also contains the realistic dataset size for each of the scenarios.

It is also important to evaluate the performance, i.e., execution time and memory usage, of the developed *Learning Engine*. Although, the learning process will mostly run in the background, it is important to assess the amount of resource usage. Most healthcare environments have a limited amount of resources and delegating the processing to the cloud is often difficult because of privacy issues. To evaluate the influence of noise on the performance, the same datasets were used as for the correctness tests. However, to assess the influence of the size of the dataset, datasets were generated with sizes ranging from 1,000 to 30,000 in steps of 1,000 instances. Bigger datasets were used as it is important to explore the limits of the proposed self-learning components.

To achieve reliable results, each test was repeated 35 times, of which the first three and the last two were omitted during processing. For each run, a new dataset was generated. Finally, the averages across the 30 remaining runs are calculated and visualized in the form of graphs. The tests were performed on a computer with the following specifications: 4096 megabyte (MB) (2 x 2048 MB) 1067 megahertz (MHz) Double Data Rate Type Three Synchronous Dynamic Random Access Memory (DDR3 SDRAM) and an Intel Core i5-430 Central Processing Unit (CPU) (2 cores, 4 threads, 2.26 gigahertz (GHz), 3 MB cache).

6.5.1 Generating the dataset to discover thresholds

As indicated in Section 6.4.1, the dataset consists of seven input attributes, i.e., the probabilistic assignment of a call to the priority classes. As label, the assigned priority of the call is used. The dataset is generated in such a way that discovered thresholds should be the ones that are currently being used by the oNCS, i.e., 0.21 - 0.3 - 0.24 - 0 - 0.05 - 0 - 0, ordered from the highest to the lowest priority.

To generate a new instance of the dataset, a priority label is first chosen. The label is chosen such that the distribution of the generated calls amongst the different priority classes reflects the following realistic distribution determined by domain experts: 5% - 10% - 25% - 35% - 25% - 0% - 0%, ordered from the highest to the lowest priority. Based on this label, the probabilistic values for the input attributes are generated. For all the priority classes that are checked earlier by the threshold algorithm than the assigned priority, a probabilistic value is randomly generated that is smaller than the threshold for this priority. For example, if a call with a High priority is being created, then the probabilistic value for the Highest priority will be lower than 0.21. For the assigned priority, a random probabilistic value is generated, which is higher than its threshold. Finally, for the remaining priority classes, a random probabilistic value is generated. The thresholds for these priorities are thus not taken into account.

To introduce noise in the generated datasets, the priority labels of some generated instances are changed. This means that they receive a different label than the one which would be assigned by the threshold algorithm and which was used to generate these instances. For a noise percentage of x, each generated instance has $x\%$ chance of being assigned a priority label that is one level higher or one level lower than the correct priority label. Some generated instances are shown in Table 6.2. The labels indicated in italic represent noise.

6.5.2 Generating the dataset to discover the probabilities for the priorities

The dataset generated for this scenario contains two input attributes, i.e., the type of the call and the risk group of the patient who made it, and the assigned priority as label. To create a new instance, a risk group is randomly assigned based on the following distribution: 20%, 50% and 30% chance of being a High, Medium or Low Risk patient respectively. Moreover, the instance has 60%, 30% and 10% chance of being a Normal, Sanitary and Service call respectively. These distributions were determined based on input from domain experts. Using the parameters already defined in the oNCs and visualized in Table 6.1, the probabilistic assignment of this generated call to the various priority categories is determined. For example, if an instance is generated with the input attributes Normal type of call and High Riks patient, then it has 20%, 60% and 20% chance of receiving the High, Above Normal and Normal priorities respectively. Based on this distribution, a priority is randomly chosen as label.

Similar as in the previous scenario, noise is introduced by changing the label of an instance to a priority that is one lever higher or lower than the assigned one. Some generated instances are shown in Table 6.3. The labels indicated in italic represent noise.

6.6 Results & discussion

6.6.1 Correctness of the discovered thresholds

To assess the correctness, the relative error of the discovered thresholds is calculated. The relative error expresses how much the learned threshold deviates from the threshold on which the dataset generation was based. For example, a relative error of 5% for the threshold of the Above Normal priority indicates that the discovered threshold deviates at most 5% from 0.24. The oNCS employs a threshold of 0 for the Normal, Low and Lowest priority categories to ensure that the default priority assigned to calls is the Normal priority. The Low and Lowest priorities are generally reserved for particular types of calls, e.g., technical assistance calls. Because of the way the dataset generation algorithm takes these zero thresholds into account to generate the instances, these thresholds are always discovered. Therefore, only the other, non-zero, thresholds are discussed.

Figure 6.8 depicts the relative error of the discovered thresholds as a function of the dataset size. It can be derived that very accurate thresholds are obtained, even when datasets with a small amount of instances are used. When the dataset contains at least 500 instances, the relative error stays smaller then 0.5% for all the thresholds. As mentioned previously, on average five calls are launched per patient in a department with on average 30 patients. Consequently, four days after

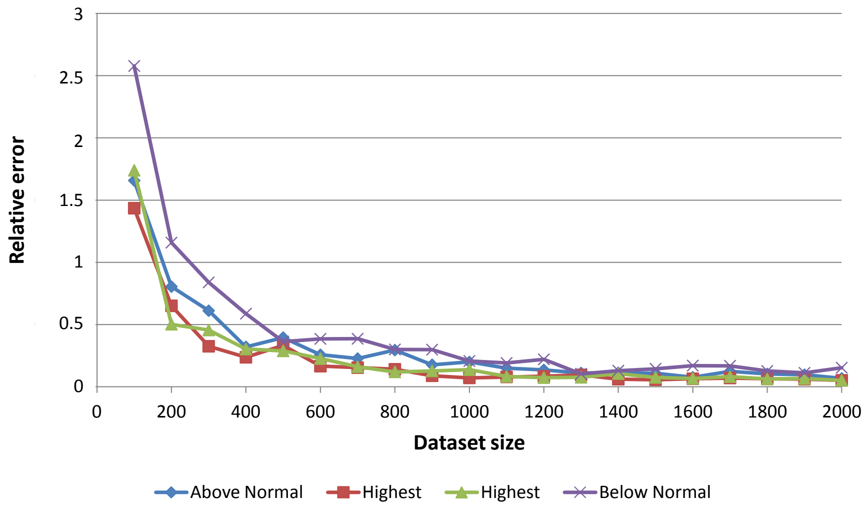


Figure 6.8: The relative errors of the thresholds discovered for the different priority categories as a function of the size of the dataset

deployment of the oNCS enough data would be collected to accurately adjust the thresholds to the behavior of the caregivers. Note that for small datasets, more accurate results are obtained for the thresholds of higher priority classes. A separate decision tree is built for each priority class, based on a subset of the total dataset. In these subsets the instances are removed, which received as label a higher priority class than the one that the decision tree is currently being built for. Consequently, the decision trees for lower priorities are trained on less data than the decision trees for higher priorities. As a result, these lower priorities exhibit a higher relative error for small datasets.

Figure 6.9 visualizes the relative errors for the discovered thresholds as a function of the amount of noise in a realisticly sized dataset of 1,050 instances. It is clear that the *Learning Pipeline* is insensitive to a noise rate of less than 20%, as they result in relative errors for the thresholds of less than 5%. If the amount of noise increases beyond this point, the relative errors quickly rise to 10% and higher. The relative error of the threshold of the Below Normal priority is higher than the ones of the Normal and High priority because it is trained on smaller datasets, as explained in the previous paragraph. The relative error of the threshold of the Highest priority is much higher than the others. This is the first threshold that needs to be determined. Consequently, it is trained on a dataset with a very high amount of instances labeled as “Other”. This skewed dataset, containing more negative than positive examples, results in a higher relative error for this priority.

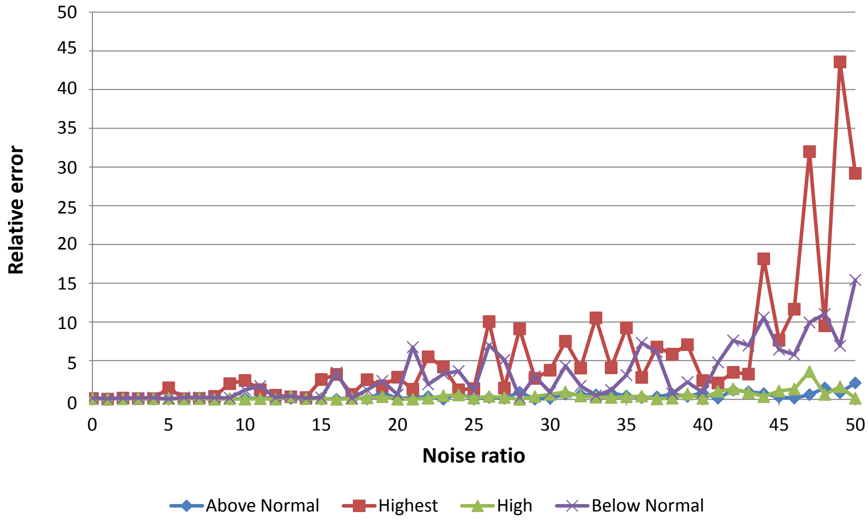


Figure 6.9: The relative errors of the thresholds discovered for the different priority categories as a function of the amount of noise in the dataset

6.6.2 Correctness of the discovered probabilities

The dataset for this scenario consists of two input attributes, namely the risk group of the patient and the type of the call, which each can have three possible values. The priority label can have seven possible values. Consequently the Bayesian network needs to determine 63 probability parameters. It is difficult to give a clear overview of all the calculated parameter values for all the different dataset sizes and noise ratios. Therefore, Table 6.5 visualizes only the relative errors for the discovered probabilities for a dataset of realistic size, i.e., 1,050 instances, without noise. Despite the large number of parameter values that need to be deduced from a relatively small dataset, the relative errors are quite small. Three discovered probabilities have a relative error bigger than 10%. These errors are indicated in italic in Table 6.5. However, all the other derived parameter values deviate only on average 3% and maximum 6% from the correct value. It can also be noted that higher relative errors correspond to situations that do not occur often in reality. As the dataset is generated based on realistic distributions, these situations are represented by less instances in the dataset. This makes it more difficult for the Bayesian network to obtain a correct parameter value for these situations. For example, as explained in Section 6.5.2, an instance only has 10% chance to receive the type Service and 20% chance of being launched by a High Risk patient. Consequently, there's only 2% chance that an instance is generated that fulfills both of

Risk group	Type of call	Relative error						
		Highest	High	Above normal	Normal	Below normal	Low	Lowest
High	Normal		0.01	0.03	0.01			
	Sanitary		0.06	0.02	0.05			
	Service			0.04	0.04	*0.16*		
Medium	Normal			0.00	0.04	0.02		
	Sanitary			0.04	0.04	0.01		
	Service				0.02	0.05	*0.14*	
Low	Normal				0.06	0.03	0.03	
	Sanitary				0.01	0.02	0.02	
	Service					0.03	0.02	*0.12*

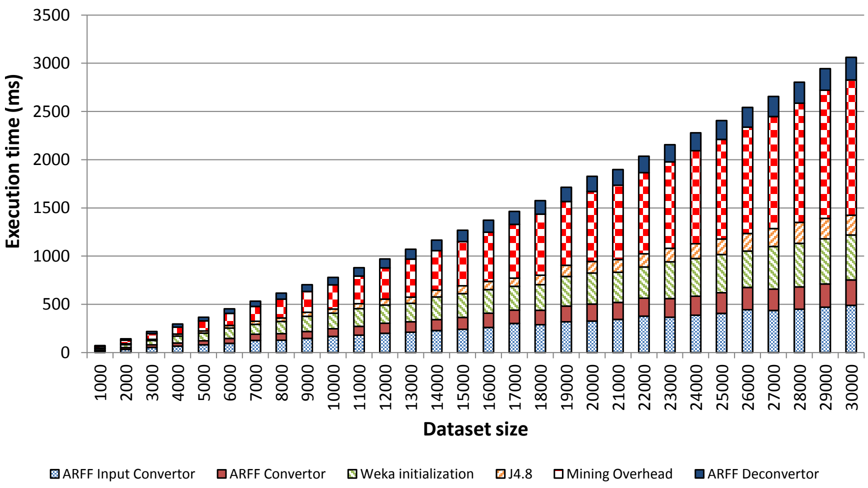
Table 6.5: Relative error for the discovered probability parameters for a dataset with 1,050 instances

these criteria. As a result, the relative error for this probabilistic value is 0.16%.

6.6.3 Execution time of the threshold *Learning Pipeline*

The execution time as a function of the size of the dataset is depicted in Figure 6.10. The execution times of the *Threshold Extractor*, *Counter Reliability Algorithm* and *Threshold Filter Algorithm* are negligible compared to the execution times of the visualized components. The execution time of the *Priority Threshold Integration Component* depends heavily on the complexity and the amount of data in the ontology as this component checks the consistency of the ontology after the parameters are adjusted. As the ontology was not initialized with a realistic data set, e.g., representing a realistic amount of staff members and patients, the execution time of this module is not shown. The processing of the data by the *Iterative Tree Miner* can be split up into three parts. The *Mining Overhead* denotes the time needed to pre-process the dataset such that the different decision trees can be built as explained in Section 6.4.2.1. The *Weka Initialization* step consists of transforming the ARFF format to Java Objects, while *J4.8* algorithm builds the actual decision tree using WEKA. The execution times of these three steps are visualized separately.

It can be derived from Figure 6.10a that the execution time is exponential as a function of the size of the dataset. Figure 6.10b shows that this is caused by the exponentially increasing execution time of the *Mining Overhead*. The execution times of the other components are linear as a function of the amount of instances. The complexity of the J4.8 algorithm is $O(m * n^2)$ for a dataset with m instances and n attributes [26]. The number of attributes is constant in this scenario, i.e., one input attribute and one label per decision tree built for a particular priority. Consequently, the complexity reduces to $O(m)$ and thus becomes linear in the number of instances. The `ARFF Input Convertor`, `ARFF Conver-`



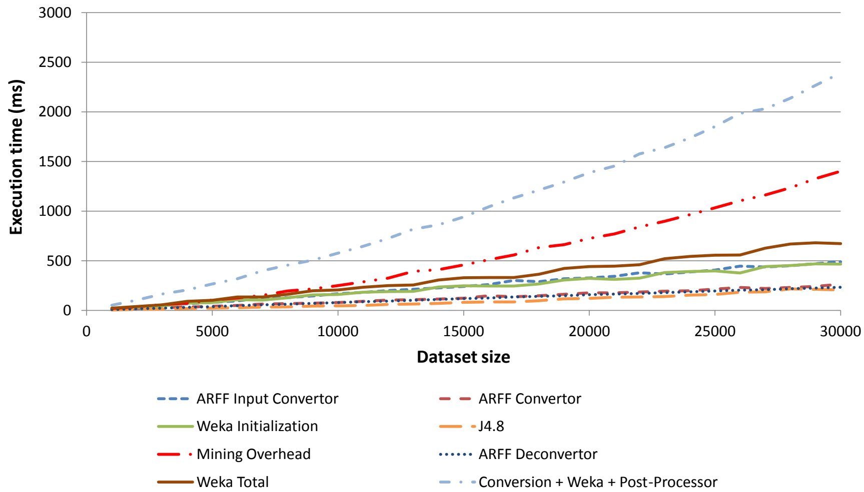


Figure 6.10: Execution time as a function of the dataset size for the different components of the threshold Learning Pipeline

tor and ARFF Deconvertor are also linear in the size of the dataset, as they need to (de)convert all the instances one by one. It can also be noted that the *ARFF Input Convertor* consumes more time than the *ARFF Convertor*. The first translates a String-based representation of the dataset, while the second receives the instances expressed in the internal data format as input. This second, structured representation can be processed more easily.

Figure 6.11 analyzes the execution time of the *Mining Overhead* in more detail. As explained in Section 6.4.2.1, a dataset is constructed for each priority by

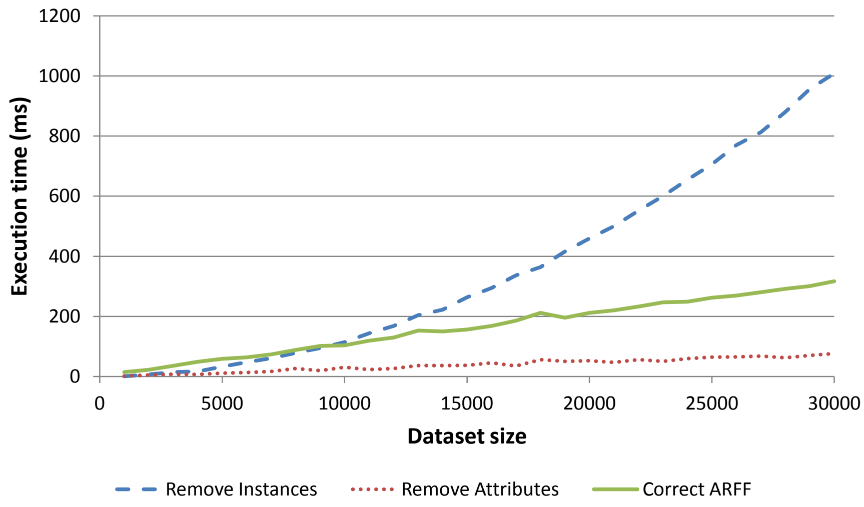


Figure 6.11: Execution time as a function of the dataset size for the different steps of the Mining Overhead

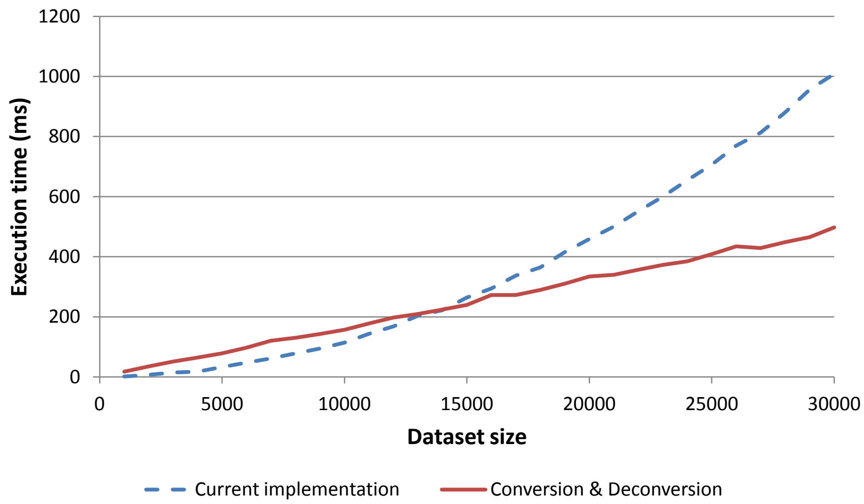


Figure 6.12: Compares the execution times of removing instances from the dataset as a function of the dataset size for the current and alternative implementation

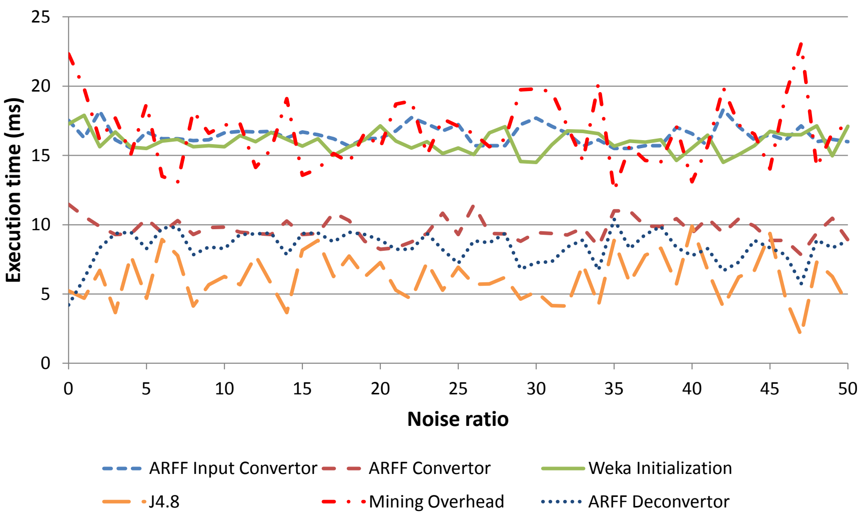
removing the input attributes related to the other priority classes, removing all the instances labeled with a higher priority and renaming all the lower priority labels as "Other". Figure 6.11 indicates that most of the execution time is consumed by removing the instances. A possible solution is removing the instances before the dataset is translated to the ARFF format. The complexity of removing instances from the dataset, represented in the internal data format, is linear in the size of the dataset. However, this solution also requires that each separate dataset is translated by the *ARFF Convertor*. This also increases the execution time as there is significant overlap between the datasets and thus more instances need to be converted. Figure 6.12 compares the execution time of the current implementation for removing the instances with the additional execution time, which is needed to (de)convert the separate datasets for the alternative solution. The additioal execution time of the alternative implementation is linear in the amount of instances. However, it only achieves a better performance for bigger datasets with at least 15,000 instances. As 1,050 instances was deemed to be a realistic size of the dataset, the current implementation is preferred.

Figure 6.13a depicts the execution time as a function of the amount of noise for the realistic dataset containing 1,050 instances. As the measured execution times are quite small, i.e., lower than 25 ms, the graphs are quite erratic and unpredictable. To get a clear view on the underlying trends, the performance tests were repeated for a dataset consisting of 5,000 instances. The resulting graph is visualized in Figure 6.13b. It can be derived that the influence of the amount of noise on the execution time is negligible. The dataset for each decision tree consists of only one input attribute and a label, which can only assume two values. Consequently, increasing the amount of noise will not have a large impact on the complexity of the constructed decision tree.

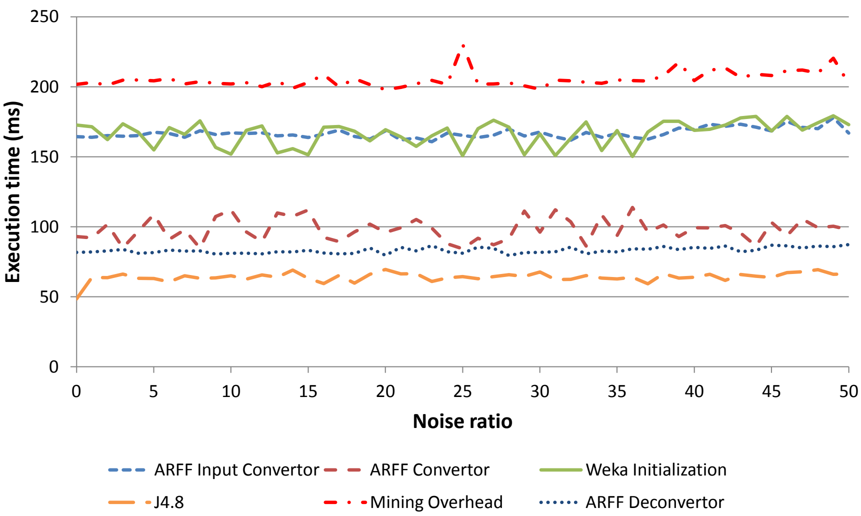
It can be concluded that a dataset with a realistic size of 1,050 instances can be processed in less than 100 ms, irrespective of the amount of noise.

6.6.4 Execution time of the probabilities *Learning Pipeline*

The execution time as a function of the size of the dataset is depicted in Figure 6.14. The execution times of the *Probability Calculator*, *Fluctuation Reliability Algorithm*, *Threshold Filter Algorithms* and *Priority Probability Integration Component* are not shown for the same reasons as in the previous section. The *Bayes Net Miner* consists of only two steps, namely initializing Weka and building the model using the *BayesNet* algorithm of Weka. The execution times for these two steps are visualized seperately. It can be noted that the the execution time is linear as a function of the size of the dataset. Figure 6.14b illustrates that the execution time of each of the individual components is also linear as a function of the size of the dataset. The execution times are also very small. The input conversion and initialization of Weka consume most of the execution time. Building the Bayesian network only requires a small amount of time, namely at most 20 ms for a dataset of 30,000 instances. The complexity of the Bayesian network is



(a) Dataset of 1,050 instances



(b) Dataset of 5,000 instances

Figure 6.13: Execution time as a function of the amount of noise in the dataset for the different components of the threshold Learning Pipeline

the same as the J4.8 algorithm, namely $O(m * n^2)$ for a dataset with m instances and n attributes [27]. As the amount of attributes does not change in this scenario, this complexity also reduces to $O(m)$ and thus becomes linear in the number of instances. The difference in execution time between the ARFF Input Convertor and ARFF convertor was already explained in the previous section.

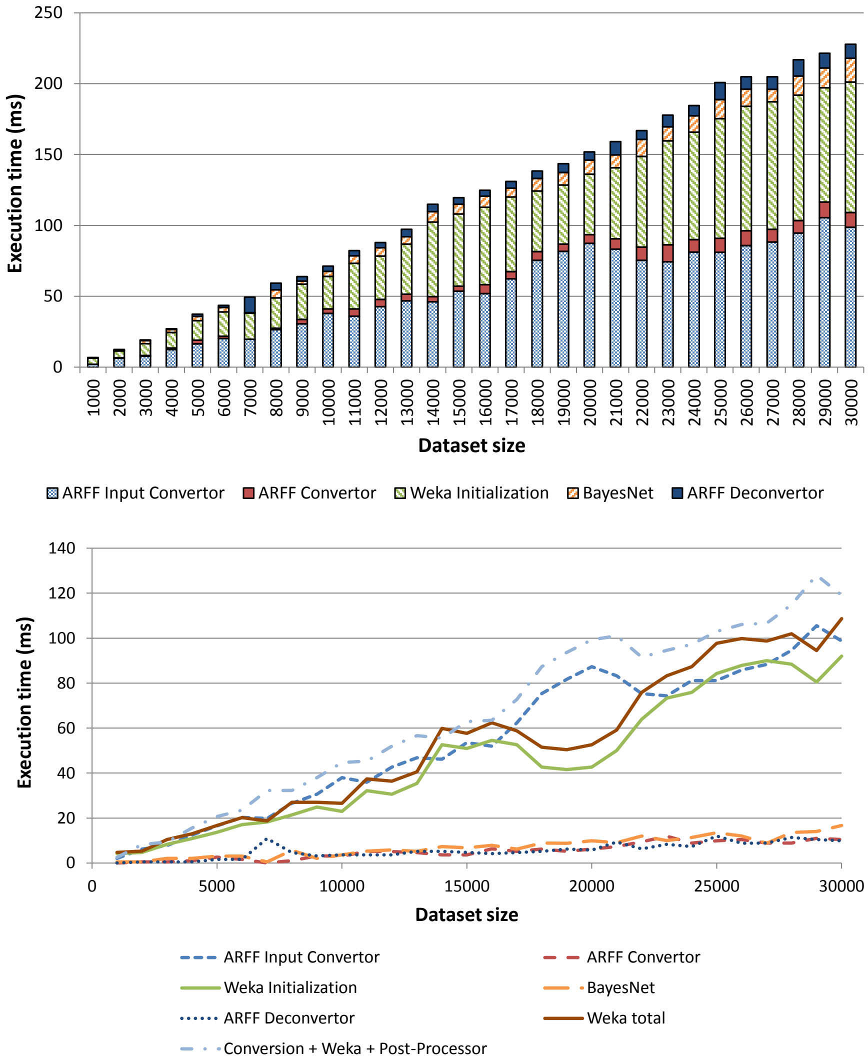
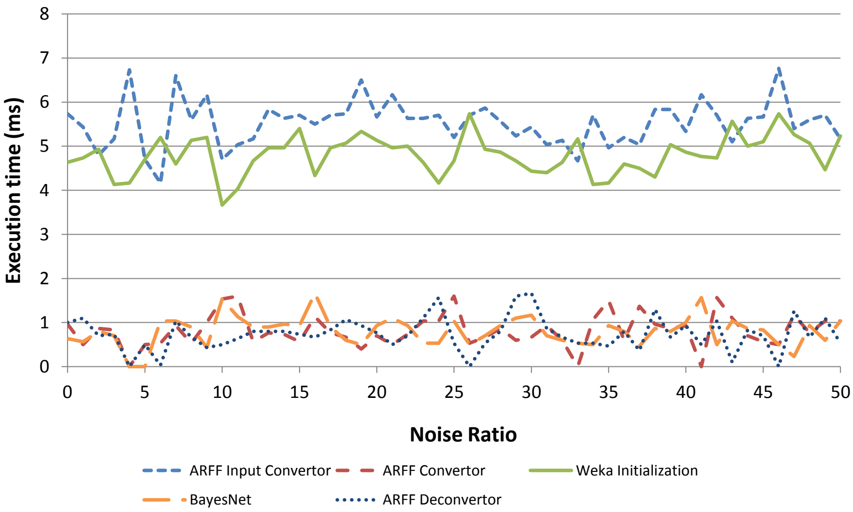
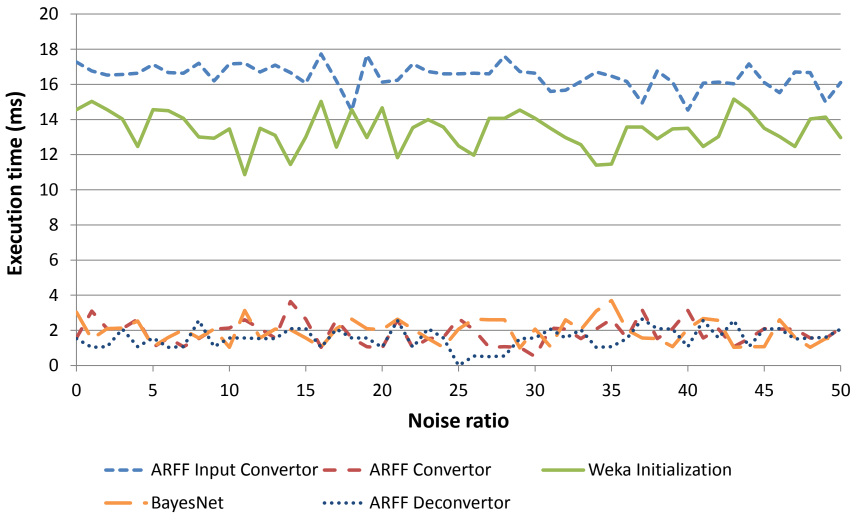


Figure 6.14: Execution time as a function of the dataset size for the different components of the probabilities Learning Pipeline

Figure 6.15a depicts the execution time as a function of the amount of noise for the realistic dataset containing 1,050 instances. Again, these execution times are too small, i.e., lower than 7 ms, to perceive a clear trend and the tests were repeated for a dataset of 5,000 instances, as shown in Figure 6.15b. Similarly to the previous section, it can be concluded that the influence of the amount of noise on the execution time is negligible.



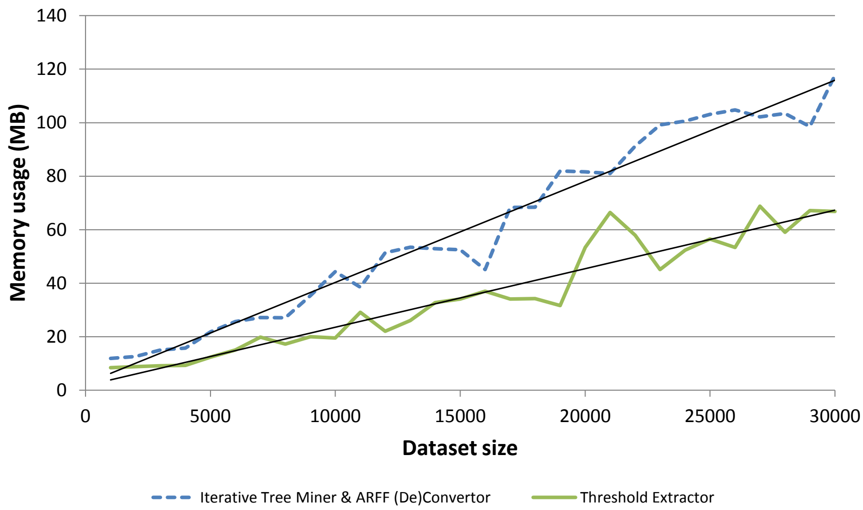
(a) Dataset of 1,050 instances



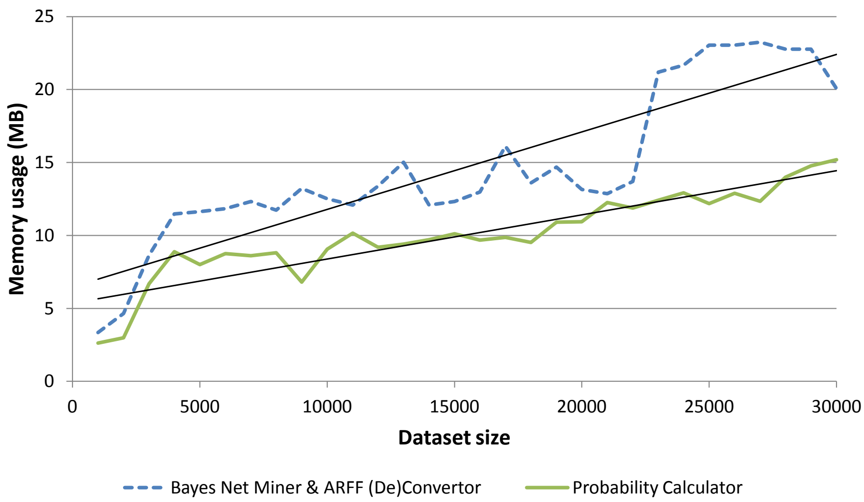
(b) Dataset of 5,000 instances

Figure 6.15: Execution time as a function of the amount of noise in the dataset for the different components of the probabilities Learning Pipeline

For this scenario, it can also be concluded that, irrespective of the amount of noise, the execution time is very good and negligible for datasets of a realistic size of 1,050 instances, i.e., less than 20 ms.



(a) Threshold *Learning Pipeline*



(b) Probabilities *Learning Pipeline*

Figure 6.16: The memory usage as a function of the size of the dataset

6.6.5 Memory usage

Figure 6.16 illustrates the memory usage of the Learning Pipeline for both scenarios as a function of the size of the dataset. The fluctuating pattern of the graphs can be explained by the memory that is consumed by the *Garbage Collector* in Java. However, trend lines can clearly be discerned. It can be noted that the memory usage is linear as a function of the amount of instances. Moreover, the total amount of consumed memory stays quite low, i.e., at most about 120 MB for the threshold *Learning Pipeline* and 25 MB for the probabilities scenario. For the realistic dataset of 1,050 instances, the memory usage is negligible for both scenarios, namely lower than 5 MB for the probabilities *Learning Pipeline* and 20 MB for the threshold scenario. The memory usage for the threshold scenario is significanlty higher. This can be explained by the different datasets that need to be created and stored to build the decision trees for each of the priorities.

6.7 Conclusion

This article describes our experiences with extending the oNCS with self-learning components such that it can automatically adjust its parameters. This ensures that the application is tuned towards the needs and requirements of the caregivers and increases its adoption. Moreover, caregivers are no longer burdened with trying to define accurate parameter values for the application at development time or tweak its configuration at run-time.

The self-learning extension consists of the following steps. First, *Monitoring Algorithms* are used to monitor how the application is used with a certain context. These algorithms gather and store data. When enough data has been collected the *Data Collection Component* and *Input Convertor* retrieve the data and transform it to the internal data format used by the self-learning components. Second, the *Pre-Processor* cleans the data. *Data Mining* techniques and a *Post-Processor* are used to discover the new parameter values. The *Decision Component* associates probabilities with these learned parameter values to express their reliability. Values with a too low probability are filtered. Finally, the *Integration Component* integrates the new parameter values and their associated reliability in the oNCS.

The oNCS contains two types of parameters, namely thresholds and probabilities. An extensive evaluation was performed to assess the applicability, correctness and performance of the self-learning components for both scenarios. For the thresholds, it was shown that correct results with a relative error of less than 5% are obtained when the dataset contains at least 500 instances, i.e., calls, and the noise ratio is less than 20%. For the probabilities, it was deduced that for a realistic dataset of 1,050 instances correct results were obtained. Both the threshold and probability parameters are learned very efficiently as the components require at most 100 ms execution time and 20 MB memory for a realistic dataset of 1,050 instances, irrespective of the amount of noise in this dataset.

Future work will mainly focus on evaluating a prototype of the self-learning oNCS in a real-life setting.

	Correctness			Execution time as a function of		Realistic dataset		
	Relative error	**Dataset size**	**Amount of noise**	**Dataset size**	**Amount of noise**	**Size**	**Execution time**	**Memory usage**
Chapter 5	$< 1\%$	$> 1{,}000$ instances	$< 5\%$	Linear	Decreases as noise increases	1,680 instances	< 100 ms	< 10 MB
Chapter 6 thresholds	$< 0.5\%$	> 500 instances	$< 20\%$	Exponential	negligible	1,050 instances	< 100 ms	< 20 MB
Chapter 6 probabilities	/	/	/	Linear	negligible	1,050 instances	< 20 ms	< 5 MB

Table 6.6: Comparing the evaluation results of Chapter 5 and Chapter 6.

6.8 Addendum

Some additional remarks are highlighted in this Addendum.

First, it can be noted that a *Learning Ontology* was presented in Chapter 5, which allows associating the learned knowledge with its origin. This ontology can also be used to indicate how the *Learning Pipeline* was configurated to learn the parameters of the oNCS. This is illustrated for a learned threshold in Figure 6.17.

Second, Table 6.6 compares the evaluation results of Chapter 5 and Chapter-chap:selfLearningONCS.

Third, some performance loss can be noted when the performance of the self-learning oNCS is compared to the performance of the original oNCS presented in Chapter 4. This is due to the fact that the self-learning framework adds additional concepts, axioms and probabilities to the ontology of the oNCS. As mentioned previously and thoroughly discussed in Chapter B, the execution time of the probabilistic Reasoner Pronto increases as the amount of probabilistic statements increases. Pronto can only handle up to 15 probabilistic statements in a performant manner. Moreover, the larger amount of axioms also makes the deterministic reasoning with Pellet more complex.

Finally, it can be noted that the sequential learning pipeline of the self-learning framework is vulnerable to Single Points of Failure. To counter this, the intermediate results can easily be saved using the internal data format such that not all work is lost when one component of the pipeline fails. Moreover, as the different pipes are interchangeable, they can easily be duplicated to make the self-learning framework more robust.

Acknowledgment

F. Ongenae and M. Claeys would like to thank the IWT for financial support through their Ph.D. grant.

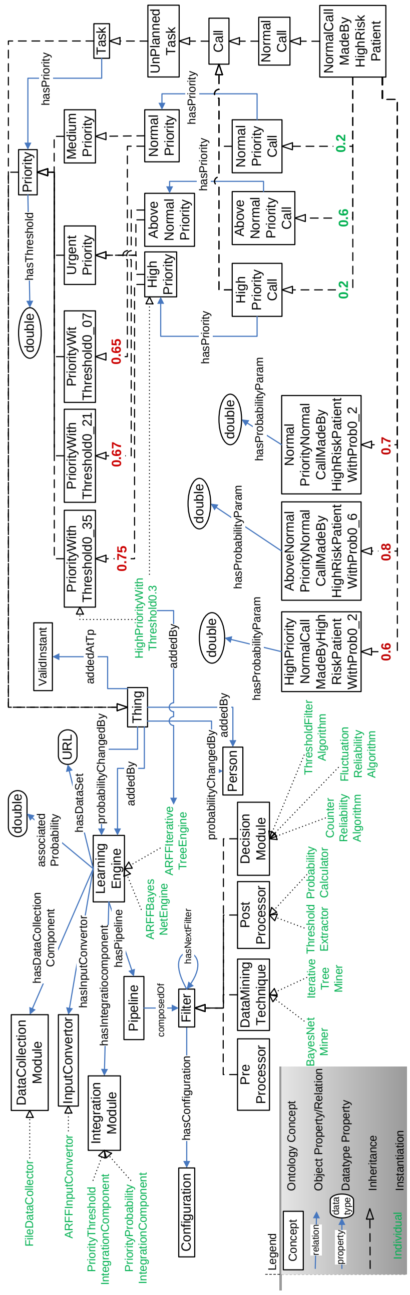


Figure 6.17: Conveying knowledge about the adjusted parameters by using the Learning Ontology

References

[1] World Health Organization (WHO). *Health topics: Ageing*. http://www.who.int/topics/ageing/en/, 2013.

[2] I. Meyer, S. Müller, L. Kubitschke1, A. Dobrev, R. Hammerschmidt, W. B. Korte1, T. Hüsing, T. van Kleef, S. Otto, J. Heywood, and M. Wrede. *eCare as a way of coping with an ageing population today and tomorrow. The eCare Benchmarking study*. Technical report, European Commission, Directorate General Information Society and Media, Brussels, April 12 2013. http://ec.europa.eu/information_society/newsroom/cf/itemdetail.cfm?item_id=10182.

[3] World Health Organization (WHO). *The World Health Report 2006 - working together for health*. http://www.who.int/whr/2006/en/, 2006.

[4] E. Percy. *Healthcare Challenges and Trends*. Technical report, Logica, 2012.

[5] C. Orwat, A. Graefe, and T. Faulwasser. *Towards pervasive computing in health care - A literature review*. BMC Medical Informatics and Decsion Making, 8(26):18, 2008.

[6] J. Li, A. Talaei-Khoei, H. Seale, P. Ray, and C. R. MacIntyre. *Health Care Provider Adoption of eHealth: Systematic Literature Review*. Interactive Journal of Medical Research, 2(1):e7, 2013.

[7] J. H. Jahnke, Y. Bychkov, D. Dahlem, and L. Kawasme. *Context-aware information services for health care*. In Proc. of the Workshop on Modeling and Retrieval of Context, pages 73–84, 2004.

[8] J. Criel and L. Claeys. *A transdisciplinary study design on context-aware applications and environments. A critical view on user participation within calm computing*. Observatorio, 2(2):57–77, 2008.

[9] F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck. *An ontology-based nurse call management system (oNCS) with probabilistic priority assessment*. BMC Health Services Research, 11:26, 2011.

[10] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, A. Jacobs, M. Van Gils, A. Ackaert, S. De Zutter, P. Verhoeve, and F. De Turck. *Participatory Design of a Continuous Care Ontology: Towards a User-Driven Ontology Engineering Methodology*. In J. Filipe and J. L. G. Dietz, editors, Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, 26-29 October 2011. ScitePress Digital Library;.

[11] T. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199–220, 1993.

[12] M. Strobbe, O. V. Laere, F. Ongenae, S. Dauwe, B. Dhoedt, F. D. Turck, P. Demeester, and K. Luyten. *Novel applications integrate location and context information*. IEEE PERVASIVE COMPUTING, 11(2):64–73, 2012.

[13] S. Haiges. *A Step By Step Introduction to OSGi Programming Based on the Open Source Knopflerfish OSGi Framework*. Technical report, October 2004.

[14] D. L. McGuinness and F. v. Harmelen. *OWL Web Ontology Language Overview*. Technical Report REC-owl-features-20040210, World Wide Web Consortium, February 10 2004. http://www.w3.org/TR/owl-features/.

[15] P. Klinov. *Pronto: A Non-monotonic Probabilistic Description Logic Reasoner*. In Proceedings of the 5th European Semantic Web Conference, pages 822–826, Tenerife, Spain, June 1-5 2008.

[16] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. *Jena: implementing the semantic web recommendations*. In Proceedings of the 13th international conference on World Wide Web, Alternate track papers & posters, pages 74–83, New York, NY, USA, May 17–22 2004.

[17] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 2nd edition, 2003.

[18] I. H. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan-Kaufmann, 3rd edition, 2011.

[19] E. Prud'hommeaux and A. Seaborne. *SPARQL Query Language for RDF*. W3C Recommendation REC-rdf-sparql-query-20080115, January 15 2008. http://www.w3.org/TR/rdf-sparql-query/.

[20] S. B. Kotsiantis. *Decision trees: a recent overview*. Artificial Intelligence Review, 39(4):261–283, 2013.

[21] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice-Hall, San Francisco, CA, USA, 2003.

[22] S. B. Kotsiantis. *Supervised Machine Learning: A Review of Classification Techniques*. Informatica, 31(3):249–268, 2007.

[23] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, USA, 1993.

[24] T. Lukasiewicz. *Probabilistic Description Logics for the Semantic Web*. Technical report, Technical University of Wien, Institute for Information Systems, Wien, Austria, 2007.

[25] *Ghent University hospital*. http://www.healthcarebelgium.com/index.php?id=uzgent, 2013.

[26] J. Su and H. Zhang. *A Fast Decision Tree Learning Algorithm*. In Proceedings of the 21st National Conference on Artificial Intelligence, pages 500–505, Boston, MA, USA, 2006.

[27] J. Su and H. Zhang. *Full Bayesian Network Classifiers*. In Proceedings of the 23rd International Conference on Machine Learning (ICML), pages 897–904, Pittsburgh, PA, USA, 2006.

7

Design of a Probabilistic Ontology-based Clinical Decision Support System for Classifying Temporal Patterns in the ICU: a Sepsis Case Study

F. Ongenae, T. Dhaene, F. De Turck, D. Benoit, and J. Decruyenaere.

In Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems (CBMS), Perth, Australia, Pages 389–394, October 2010.

"Declare the past, diagnose the present, foretell the future."

– Hippocrates (460 BC - 370 BC)

Medical time series contain important information about the condition of a care receiver. It is thus important to take this temporal data into account when developing healthcare applications and services. Therefore extensions of the developed O'Care Platform (see Chapter 3), continuous care ontology (see Chapter 2) and self-learning framework (see Chapter 5) are proposed to represent and reason

with medical time series. This research is thus related to Research Contributions 6 discussed in Section 1.3 of Chapter 1.
In Chapters 5 and 6 "white-box" data mining techniques were used to detect trends and patterns in the gathered historical data. Consequently, the inner logic of the results of these techniques can be inspected and translated to new knowledge, e.g., rules, that is added to the knowledge base. In this chapter, the self-learning framework is used in combination with "black-box" machine learning techniques. Only the result of the classification technique can thus be added to the knowledge base, e.g., patient has a particular disease with 90% probability. The knowledge, which was used to arrive at this conclusion cannot be added as it is unknown due to the black-box nature of the learning techniques. Consequently, instead of letting new rules added to the ontology classify the new data, the data is processed by the trained machine learning technique each time to derive the new knowledge. This chapter also contains a detailed description of how the self-learning framework presented in Chapter 5 was adapted to integrate these black-box machine learning techniques. Appendix D dives further into the investigation of machine learning techniques to classify medical time series data by presenting an elaborate evaluation of the advantages of using Echo State Networks (ESNs) instead of other traditional classifiers combined with feature extraction and selection.

⋆ ⋆ ⋆

Abstract Medical time series contain important information about the condition of a patient. However, due to the large amount of data and the staff shortage, it is difficult for physicians to monitor these time series for trends that suggest a relevant clinical deterioration due to a complication or new pathology. This paper proposes a framework that supports physicians in detecting patterns in time series. It has three main tasks. First, the time-dependent data is gathered from heterogeneous sources and the semantics are made explicit by using an ontology. Second, Machine Learning techniques detect trends in the semantic time series data that indicate that a patient has a particular pathology. However, computerized classification techniques are not *100%* accurate. Therefore, the third task consists of adding the pathology classification to the ontology with an associated probability and notifying the physician if necessary. The framework was evaluated with an ICU use case, namely detecting sepsis. Sepsis is the number one cause of death in the ICU.

7.1 Introduction

The rapid development of computing technologies has a major impact on healthcare, particularly in intensive care units (ICUs). This growth in computer technologies has been accompanied by an increase in complexity and number of monitoring equipment, thus generating large amounts of data that must be rapidly interpreted by the medical staff. Databases have become an essential part of ICUs to store, integrate and share data about the medical condition of a patient. Time series often appear in these databases and contain important information about the condition of a patient. For instance, it has already been shown that subtle changes in certain laboratory values reflecting vital organ function, even within the normal range, are of prognostic value for sepsis [1]. However, it is difficult for physicians to continuously monitor these time-dependent parameters for subtle or sometimes even overt changes that suggest a relevant clinical deterioration due to a complication or new pathology because of the large amount of data and staff shortage. This is particularly true for junior or senior physicians who lack clinical expertise in the field of the pathology. To solve this, this paper proposes a Clinical Decision Support System (CDSS) that gathers time series data about patients from various sources, detects trends and alerts the clinicians of possible new pathologies or complications.

A first challenge is that temporal knowledge is often not explicitly represented by the database. E.g., if a time point $09/04/2010$ $5{:}42{:}00$ is associated with a White Blood Cell (WBC) count, the exact meaning of this time point is not known. It could be the time at which a blood sample was drawn or the time at which it was analyzed in the laboratory. To resolve this the framework integrates the data in an ontology [2]. Ontologies are used to represent and structure knowledge about a certain domain in a formal way. This ontology is used to annotate data with their meaning and express the relationships with other data. Moreover, it allows the integration of data coming from heterogeneous sources.

Second, computerized techniques are needed to detect trends. In the field of Machine Learning (ML) two approaches for time series classification can be identified. The first approach pre-processes the time series data by extracting and selecting features [3] such that classical statistical classification techniques can be used. The second approach focuses on using techniques which can directly cope with high-dimensional non-linear temporal data. Both approaches were integrated into the framework. These methods take the medical time series data as input and return a classification label as output e.g. indicating if a patient has the pathology or not.

Using ML techniques to predict the condition of a patient, is not 100% accurate or reliable. However, the CDSS is not meant to replace the medical staff. It is essentially developed to assist junior or non-expert senior, and to lesser extent, expert senior physicians in diagnosing a patient's condition. It is therefore important to convey the accuracy or uncertainty of the prediction to the medical practitioner. Therefore, the developed framework explicitly models this uncertainty in the ontology and reasons with it.

The remainder of this paper is structured as follows. Section 7.2 introduces the

Nr.	Indicator	Parameter	Measure type
1	Fever	Temperature blood Temperature armpit Temperature rectal	Monitored Observed Monitored
2 3	Leukocytosis Leukopenia	White Blood Cell Count (WBC)	Laboratory
4	Plasma CRP $>$ 2SD above the normal value	CRP blood	Laboratory
5	Arterial Hypotension	Non-invasive Blood Pressure (NIBP) Systolic Arterial Pressure (SAP) Levophed	Monitored Monitored Prescription
6	Thrombocytopenia	Platelet Count	Laboratory

Table 7.1: Selected diagnostic criteria of sepsis

sepsis use case. Section 7.3 details the architecture of the framework. Section 7.4 focuses on the modeling details of the sepsis ontology and Rules, while Section 7.5 describes some sepsis scenarios. Section 7.6 highlights the main conclusions and future work.

7.2 Use case: Sepsis

Sepsis [1] is a severe inflammatory response, called systemic inflammatory response syndrome (SIRS), of the body to an infection. In the US, approximately $750,000$ people are diagnosed with sepsis every year, with a mortality rate of 30%. Early detection is crucial, as appropriate therapy aimed to prevent further deterioration in organ failure reduces mortality by 15% [4]. However, sepsis can only be accurately diagnosed by the presence of a positive blood culture for a known pathogen. This infection is often discovered too late. However, there is a guideline that lists 25 possible indicators of sepsis, which allows early detection [5]. Continuously monitoring these, often time-dependent, parameters is difficult, which makes it the ideal use case for the developed framework.

To perform an initial evaluation of the framework, 6 of the 25 indicators were selected, as shown in Table 7.1. It was investigated if the framework could be used to detect trends in these parameters to automatically alert physicians of patients who might have sepsis. For this, data about 650 ICU patients, of whom 342 have sepsis, was collected during three consecutive years from the ICU databases of Ghent University hospital. The parameters collected which match these indicators are also shown in Table 7.1.

7.3 Architecture Description

The architecture of the CDSS is shown in Figure 7.1. Three layers can be discerned, as detailed below.

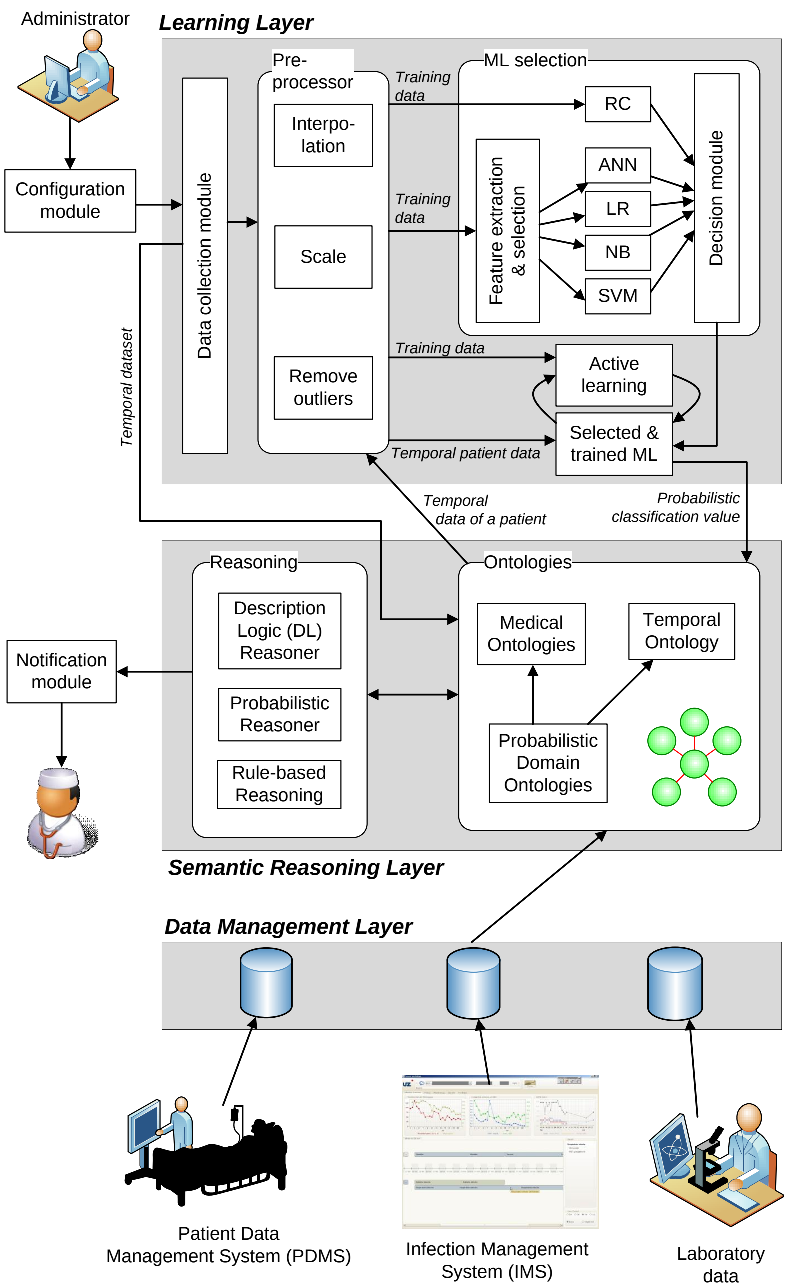


Figure 7.1: Architecture of the probabilistic ontology-based CDSS for classifying temporal patterns

7.3.1 Data Management Layer

The modern ICU contains computerized medical equipment to convey the condition of a patient such as monitoring equipment and electronic patient records integrated in a *Patient Data Management System (PDMS)*, an *Infection Management System (IMS)* which tracks the infections and antibiotic treatments of a patient and a *Laboratory* database. All this data is stored and managed in medical database systems across the hospital, as shown in the *Data Management Layer* of Figure 7.1. As mentioned previously, time series often appear in these databases and contain important information about the condition of a patient.

7.3.2 Semantic Reasoning Layer

Medical databases have some limitations, such as the lack of interoperability and standardization [6]. Moreover, as mentioned in Section 7.1, the meaning of the various time points in the database is difficult to derive. To resolve this, the *Semantic Reasoning Layer* contains *ontologies* for the integration and analysis of the medical and time series data. Existing *Medical Ontologies* are integrated into the framework and extended with *Domain Ontologies* which model the information specific to a particular hospital setting such as the measured parameter values. These *Domain Ontologies* also contain probabilistic information which express the probability that a patient has a pathology if particular conditions are met, e.g. a patient with a low WBC count has 17% chance of having sepsis. A *Temporal Ontology* associates each parameter value with its meaning in time.

Due to the foundation of ontologies in *Description Logics* (*DL*), the models can be formally proofed by using a *DL Reasoner*. This *DL Reasoner* is used to detect inconsistencies in the model as well as infer new information from the correlation of the data. E.g., a concept *Fever* is created in the ontology, which automatically detects patients with a temperature above 38 °C. More complex logic is expressed by defining *Rules* [7] on top of this ontology. These *Rules* allow expressing algorithms which take advantage of the temporal information in the ontology. This way, increasing or decreasing trends in time series are automatically detected and notified to the clinicians. This notification is handled by the *Notification Module*, which is shown on the left side of Figure 7.1. It allows to specify which clinicians are interested in what information about which patient.

7.3.3 Learning Layer

As explained in Section 7.2, not all time series trends can be easily expressed as Rules. ML techniques handle these difficult cases, as shown in the *Learning Layer* of Figure 7.1.

7.3.3.1 Data selection and pre-processing

First, training data is extracted from the ontology. Which parameters are used as input and which pathology needs to be predicted is specified by the administrator by selecting the appropriate ontology concepts in the *Configuration Module*. The *Data Collection Module* automatically extracts all the applicable data from the ontology and provides it to the *Pre-processor*.

This *Pre-processor* contains several modules to clean up the patient data. The *Remove Outliers* component removes outliers from the time series or even removes complete patients, e.g. patients for whom the input time series is too short. The *Scale* component centers the input values at zero. This is often beneficial for the learning algorithms of various ML techniques. Finally, interpolation is needed. Many parameter measurements are performed by hand, so there exists some variance in the intervals between succeeding measurements. However, the input time series needs to contain measurements over regular time intervals and these intervals must be the same for all the input parameters, so that they can be used as input for the ML techniques. Currently, the *Interpolation* component offers three techniques, namely outmiddling, linear and step interpolation.

7.3.3.2 Training the ML techniques

Two ML approaches are used to classify the pre-processed time series. The first approach extracts and selects features from the time series data such that classical classification techniques can be used. Most of these techniques have been designed with a static data model in mind and are not suitable for coping with the dynamic nature of time series. The *Feature Extraction & Selection* module first generates features from the time series such as the slope, and secondly selects the most appropriate features and thus reduces the amount of input data [3]. Next, a classifier is applied to the selected features to classify the data. Four such classifiers are integrated namely *Linear Regression (LR)*, *Artificial Neural Networks (ANN)*, *Support Vector Machines (SVM)* and *Naive Bayes (NB)* [8].

The second approach focuses on techniques which can directly cope with the temporal data such as Recurrent Neural Networks (RNN). An obstacle when using RNNs is that only a few training algorithms exist which are complex and often yield poor results [9]. However, recently an approach, called Reservoir Computing (RC) [10], was developed to simplify this training process. The key idea is to model the dynamic system producing the time series data in a reservoir consisting of a RNN. The reservoir is then read by a linear readout function. The training algorithm only affects this readout function. For training linear functions many algorithms exist such as linear regression.

Afterwards, the *Decision Module* selects the technique with the highest accuracy to be used as the classifier to detect trends for this particular pathology. This

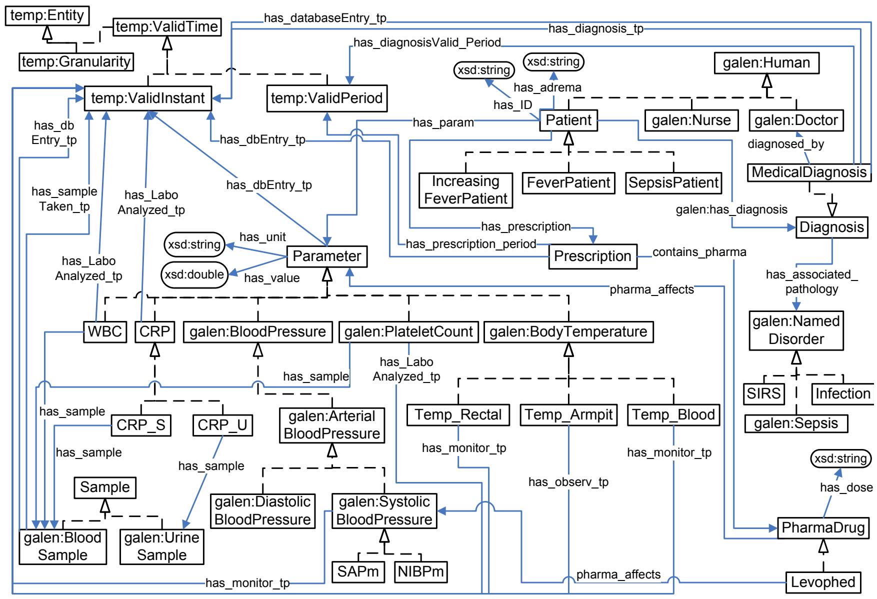


Figure 7.2: Fragment of the ontology modeling the time series of the sepsis parameters

technique is plugged into the *Selected & Trained ML* component.

7.3.3.3 Probabilistic classification and active learning

The patient, who is classified by the trained classifier, is pre-processed in the same manner as the training data. The ML technique outputs a classification label, e.g. 1 or -1, indicating if the patient has the pathology or not. It is important to convey the uncertainty of the prediction to the physician. Therefore the classification is added to the ontology with a probabilistic value that expresses its accuracy. A *Probabilistic Reasoner* is used to combine this classification with the other probabilistic data about this pathology in the ontology. This way, a probability is obtained which expresses how trust-worthy the advise of the framework is.

Finally, the *Active Learning* component continuously trains the *Selected & Trained ML* and improves its accuracy. Newly labeled data, e.g. a physician indicates that a patient has sepsis, is periodically gathered from the ontology as training data. This data is intelligently selected such that the classifier does not become overfitted, e.g. patients that were wrongly classified by the classifier are ideal input for active learning. More information can be found in [11].

7.4 Modeling the sepsis ontology and Rules

Part of the ontology that models the sepsis use case is shown in Figure 7.2. The concepts preceded with the `galen` keyword are imported from the Galen Ontology[1]. This shows nicely how other ontologies are integrated and re-used in the *Semantic Reasoning Layer*. The *SWRLTemporalOntology* [7] models the temporal information. These concepts are preceded with the `temp` keyword. By defining meaningful relationships between the medical and temporal concepts, e.g. `has_dbEntry_tp`, the semantics of the different time points in the database can be differentiated. The Protégé editor[2] was used to develop the OWL[3] ontology.

Time series data is extracted from the medical databases and automatically mapped on the ontology by using D2R[4]. D2R specifies that the `LaboTable` in the database maps on the `Parameter` concept in the ontology. This table has a column `ParamID`. Rows with `ParamID` 1500345 are mapped on the `CRP` concept. The columns `entrytime` and `datetime` are mapped on the `has_dbEntry_tp` and `has_laboAnalyzed_tp` relations respectively. The first relation expresses the time at which the data was entered into the database, while the second indicates the time point at which the analysis of the blood sample was obtained. From the `has_sample` and `has_sampleTaken_tp` relations can be derived at which time point the blood sample was drawn. Finally, the `value` column is mapped on the `has_value` relation in the ontology. This way an entire time serie is encoded in the ontology by creating multiple `CRP` instances, each associated with a value, a unit and the different time points. This `CRP` value is then associated with a `Patient` through the `has_param` relation. Note that concepts cannot only be associated with specific time points, but also with intervals during which they are valid. E.g., the `has_prescription_period` relation.

The Pellet Reasoner[5] is used as *DL Reasoner* to check the consistency and automatically classify the ontology. E.g., the concept `SepsisPatient` is defined as a patient who has `SIRS` and an `Infection`, as follows:

```
SepsisPatient
  ⊑ (∃has_diagnosis(∃has_associated_pathology SIRS)) ⊓
  ⊑ (∃has_diagnosis(∃has_associated_pathology Infection))
```

By using the SWRL Temporal Built-In Library [7], *Rules* are defined that use the temporal information in the ontology to create complex algorithms. For example,

```
FeverPatient(?p) ∧
has_bodyTemp(?p,?temp1) ∧ has_bodyTemp(?p,?temp2) ∧
has_value(?temp1,?v1) ∧ has_value(?temp2,?v2) ∧
```

[1] http://www.opengalen.org/
[2] http://protege.stanford.edu/
[3] http://www.w3.org/TR/owl2-overview/
[4] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/
[5] http://pellet.owldl.com/

```
swrlb:greaterThan(?v2,?v1) ∧
(has_observ_tp(?temp1,?t1) ∨ has_monitor_tp(?temp1,?t1)) ∧
(has_observ_tp(?temp2,?t2) ∨ has_monitor_tp(?temp2,?t2)) ∧
temp:equal(?t1,?t2, temp:Hours) ∧ temp:before(?t1,?t2) ∧
⇒
IncreasingFeverPatient(?p)
```

is a *Rule* that detects if the body temperature of a patient, who already has a fever, is still increasing within the hour. If so, the patient is categorized as an `IncreasingFeverPatient`. This allows the notification module to alert the appropriate physician.

The Reasoner Pronto[6] is used to model and reason with the probabilistic information in the ontology. Probabilistic statements are expressed in OWL by using axiom annotations. An example is discussed in Section 7.5.2.

7.5 Detailed scenario description

7.5.1 Training ML to detect sepsis

First, the *Configuration Module* is configured, as shown in Figure 7.3. The `CRP_S`, `WBC`, `NIBPm`, `SAPm`, `Levophed`, `Temp_Rectal`, `Temp_Armpit`, `Temp_Blood` and `PlateletCount` concepts are used as input and the `SepsisPatient` concept as output. The relationships used to find the associated time points are also specified, e.g. the`has_observ_tp` relation for the `Temp_Armpit` concept. The `Temp_Blood` values are preferably used as the temperature inputs as they are very reliable. If there are however `Temp_Armpit` or `Temp_Rectal` values which precede or follow the `Temp_Blood` time series, then they should be added to the temperature time series. This is configured by indicating that these parameters should be combined into 1 input parameter and specifying a ranking. Similarly, the `NIBPm` and `SAPm` parameters are combined into 1 `Bloodpressure` time series, for which `NIBPm` has the highest ranking. Finally, the period of time over which the time series data should be searched is specified.

By using these configuration settings, the *Data Collection Manager* retrieves all the patients from the ontology that have been explicitly stated as either being or not being a `SepsisPatient` by a physician by using SPARQL queries[7]. In total 650 patient with five input parameters, namely `CRP`, `WBC`, `PlateletCount`, `BloodPressure` and `Temp`, and 1 output parameter were collected.

Second, the data is pre-processed. Outliers are removed from the time series. Patients for which the time series do not have an overlapping minimal length of 40 values are also removed. This leaves 125 patients. Next, the time series are scaled to the $[-0.9, 0.9]$ interval. These bounds are chosen instead of -1 and 1 to avoid

[6]http://pellet.owldl.com/pronto
[7]http://www.w3.org/TR/rdf-sparql-query/

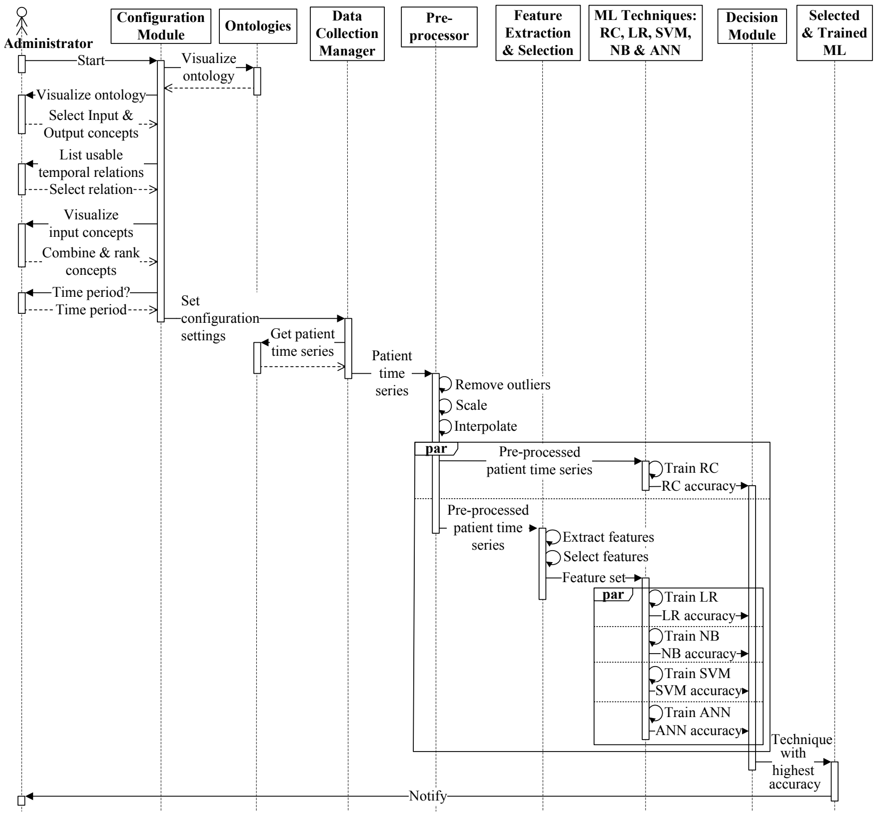


Figure 7.3: Sequence diagram of training the ML techniques

excessive weight saturation in the RC. Finally, linear interpolation is used, such that all the time series have a length of 72 time points.

Third, feature extraction and selection is performed. The following techniques are implemented in the *Feature Extraction & Selection* module to extract features from the time series: calculating the minimum, maximum, mean, median, 25^{th} and 75^{th} percentile, standard deviation, the linear regression coefficients and the area under the curve (AUC) of the time series. For the sepsis case, expert opinion reveals that the tail of the time series contains more information than the start of the series. Therefore the feature extraction was repeated for the time series each time reduced with one at the head. Finally, the slope was calculated for each two succeeding points in the time series. The slope was also calculated for each two points with a distance of 2, 3, and so further until 71. Out of all these features, useful ones are selected. Currently one feature selection technique is implemented, namely a greedy algorithm which iteratively adds the feature that improves prediction the best. This approach is similar to the one used in [12], but

Classification technique	AUC	Max. Acc.
SVM	0.712	0.700
NB	0.752	0.718
LR	0.573	0.604
ANN	0.569	0.621
RC	0.653	0.639

Table 7.2: AUC and Max. Acc. for the various ML techniques for the sepsis use case

in each iteration the set of candidates is filtered so that it contains only features that are not collinear with the already selected set. This drastically reduces the number of candidate features in each iteration and speeds up the selection process. The classifier used in this hybrid filter-wrapper [3] is the NB.

Finally, the data is used by the ML techniques as training data. The *NB* and *LR* are custom implementations, while for the *SVM*, *ANN* and *RC*, the libSVM[8], FANN[9] and RCToolbox[10] libraries were integrated. All these classifiers are trained using cross-validation and their parameters, such as the reservoir size of the *RC*, are optimized using parameter sweeps. In the current implementation, the classifiers can be optimized towards two accuracy measures, namely the AUC and the maximum balanced accuracy (Max. Acc.).

Table 7.2 shows the AUC and Max. Acc. achieved for the sepsis case with the five input time series for each of the classifiers. The *NB* classifier with *Feature Extraction & Selection* has the highest accuracy and is used as classifier.

7.5.2 Detecting a patient with sepsis

As shown in Figure 7.4, when a new patient is added to the ontology who has the required time series data for the sepsis case, this patient is automatically classified by the trained *NB*. The time series data is extracted from the ontology using SPARQL and is pre-processed in the same way as the training data. The *NB* with *Feature Extraction & Selection* returns a classification label, namely 1 or -1, indicating if the patient has sepsis or not.

To convey the accuracy of the prediction to the physician, the classification is added to the ontology as an instance axiom annotated with a probability. The annotated axiom

```
<owl11:Axiom>
    <rdf:subject rdf:resource="#patient1"/>
    <rdf:predicate rdf:resource="rdf:type"/>
    <rdf:object rdf:resource="#SepsisPatient"/>
    <pronto:certainty >0.752;0.752</pronto:certainty>
</owl11:Axiom>
```

[8] http://www.csie.ntu.edu.tw/~cjlin/libsvm
[9] http://leenissen.dk/fann/
[10] http://snn.elis.ugent.be/rctoolbox

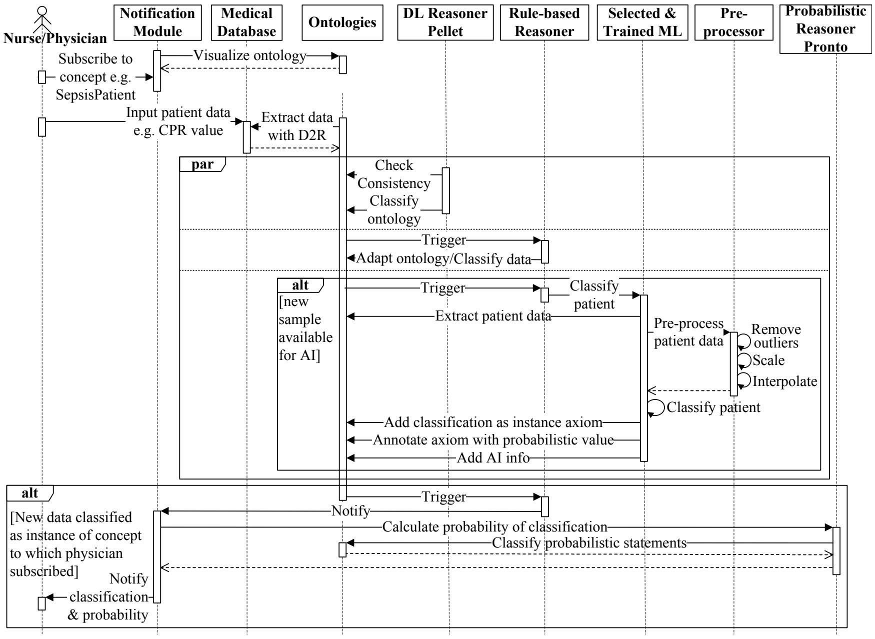


Figure 7.4: Sequence diagram of detecting a patient with sepsis

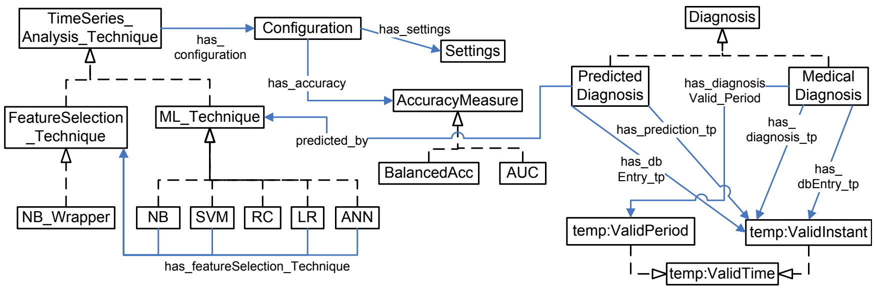


Figure 7.5: Fragment of the ontology modeling the time series classification methods

expresses that a particular patient, namely `patient1`, has 0.752% chance of being a `SepsisPatient`. The AUC of the *NB* is used as probabilistic value.

Information about the classification technique is added to the ontology, by using the concepts shown in Figure 7.5. First, an instance of `PredictedDiagnosis` is created to indicate that this classification is a predicted diagnosis. This instance is connected to the classification class `SepsisPatient` with the `has_diagnosis` relation and associated with the time point at which the prediction was made. Next, a new instance of the `NB` concept is created to specify by which `ML_technique` the classification was predicted. This `NB` instance is attached to the classification by using the `predicted_by` relation and associated with `Configuration` and `Settings` instances. These instances contain all the parameter values of the *NB*. Finally, instances of the `BalancedAcc` and `AUC` classes are created containing the values of Table 7.2 of the *NB*.

The *Notification Module* allows staff members to subscribe to classes, e.g. `SepsisPatient`, defined in the ontology. Each time an instance is classified as a member of this class, the staff member is alerted. The associated trust in this prediction is also calculated by combining all the probabilistic information in the ontology.

7.6 Conclusions

In this paper a framework was proposed to 1) augment the time series data in medical databases with semantic information by using ontologies, 2) use Machine Learning techniques to automatically classify this semantic time series data as suggesting a relevant clinical deterioration due to a complication or new pathology or not, and 3) add this classification to the ontology, while expressing the uncertainty that is associated with this prediction. Future work will mainly focus on integrating ensembles [13]. A clinical evaluation of the sepsis use case will also be performed.

Acknowledgment

Femke Ongenae would like to thank the IWT, Institute for the promotion of Innovation through Science and Technology in Flanders, for supporting this work through her PhD grant.

References

[1] J. F. Dhainaut, A. F. Shorr, W. L. Macias, M. J. Kollef, M. Levi, and K. R. en D. R. Nelson. *Dynamic evolution of coagulopathy in the first day of severe sepsis: relationship with mortality and organ failure*. Critical Care Medicine, 33(7):450–452, 2005.

[2] T. Gruber. *A Translation Approach to Portable Ontology Specifications*. Journal of Knowledge Acquisition, 5(2):199–220, April 1993.

[3] I. Guyon and A. Elisseeff. *An Introduction to Variable and Feature Selection*. Journal of Machine Learning Research, 3:1157–1182, 2003.

[4] E. Rivers, B. Nguyen, S. Havstad, J. Ressler, A. Muzzin, B. Knoblich, E. Peterson, and M. Tomlanovich. *Early goal-direct therapy in the treatment of severe sepsis and septic shock*. New England Journal of Medicine, 345(19):1368–1377, 2001.

[5] M. M. Levy, M. P. Fink, J. C. Marshall, E. Abraham, D. C. Angus, D. Cook, J. Cohen, S. M. Opal, J. Vincent, and G. Ramsay. *2001 SCCM/ESICM/ACCP/ATS/SIS International Sepsiss Definitions Conference*. Intensive Care Medicine, 29:530–538, 2003.

[6] H. V. D. Linden, S. Diepen, G. Boers, H. Tange, and J. Talon. *Towards a Generic Connection of EHR and DSS*. In Proceedings of the 19th International Congress of the European Federation for Medical Informatics (MIE2005), pages 211–216, Geneva, Switzerland, 2005.

[7] M. J. O'Connor and A. K. Das. *A Lightweight Model for Representing and Reasoning with Temporal Information in Biomedical Ontologies*. In International Conference on Health Informatics (HEALTHINF), Valencia, Spain, 2010.

[8] S. J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, USA, 3rd edition, 2009.

[9] H. Jaeger. *A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF, and the "echo state network"*. Technical Report GMD 159, German National Research Institute for Computer Science, 2002.

[10] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. *An experimental unification of reservoir computing methods*. Neural Networks, 20(3):414–423, 2007.

[11] B. Longstaff, S. Reddy, and D. Estrin. *Improving Activity Classification for Health Applications on Mobile Devices using Active and Semi-Supervised Learning*. In Proceedings of the 4th International Conference on Pervasive Computing Technologies for healthcare, 2010.

[12] P. Langley and S. Sage. *Induction of selective Bayesian classifiers*. In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94), pages 399–406, San Mateo, 1994.

[13] M. C. Lee, L. Boroczky, K. Sungur-Stasik, A. D. Cann, A. C. Borczuk, S. M. Kawut, and C. A. Powell. *A two-step approach for feature selection and classifier ensemble construction in computer-aided diagnosis*. In Proceedings of the 21st IEEE International Symposium on Computer-Based Medical Systems (CBMS), pages 548–553, Jyväskylä, Finland, 17-19 June 2008.

8
Conclusions and Research Perspectives

"The only way of discovering the limits of the possible is to venture a little way past them into the impossible."

– Arthur C. Clarke (1917 - 2008)

8.1 Final conclusions

It has been recognized that ontology-based systems can be used to improve the management of complex healthcare processes and optimize the delivery of continuous care through context-aware and pervasive services. However, the adoption of context-aware applications in the healthcare domain is lagging behind. This is due to the lack of real personalization of the services, requiring that caregivers severely change their daily work practices to accomodate the technology instead of the other way around. Moreover, a central knowledge component is often employed on which the different applications are built. As the amount of generated healthcare data is vast, this central approach has a negative impact on the performance and scalability of the platform. Finally, although time series contain important information about the condition of a patient, this time-dependent data is often not taken into account in current context-aware healthcare systems.

To address these issues, this dissertation explored the design and development of a context-aware, semantic and self-learning framework and accompanying methodologies and algorithms, which allow the user-driven development of pervasive healthcare applications that support caregivers and care receivers in their daily activities and tasks. As such, it contributes to the research fields of health informatics, semantics and knowledge management and discovery. The following sections highlight how the different research challenges discussed in Chapter 1 were tackled to reach this goal.

8.1.1 Research challenge 1 & 2: How to enable user-driven development of knowledge models without requiring IT knowledge or a lot of effort from the domain experts? & How to semantically model the exchanged continuous care data?

To tackle the first research challenge, a participatory ontology engineering methodology was presented in Chapter 2. The methodology consists of observations and five types of workshops, which actively involve ontology engineers, social scientists and domain experts. Detailed guidelines were constructed to indicate how the workshops can ideally be organized. A continuous care ontology was developed using the proposed methodology to evaluate it and simultaneously address research challenge two. The oNCS prototype was used to demonstrate to users how the healthcare knowledge was caputere in the continuous care ontology and how it used to optimize care processes.

Using the methodology, a continuous care ontology was achieved, which was tuned towards the daily work practices of the users. By employing this ontology, intelligent applications can be built that are personalized towards the needs and preferences of the caregivers and care receivers. The involved users were generally positive about the developed technology and how it took the context into account. We believe that this will increase the adoption of the technology, which is developed based on the knowledge component. Moreover, by involving the users in the development process, we noticed that some users took on the role of advocates within the organization they were affiliated with. They actively rooted for the project and the applications we were building. We feel that the availability of such advocates increases the acceptance of the new technology and its adoption. The methodology is ideally tuned towards less IT-focused domains as the workshops only require the users to role-play or brainstorm. They never have to be confronted with IT. Most involved stakeholders also only had a vague notion of what "ontology" means and were only aware that we were trying to capture their knowledge in a way that was understandable for a computer. Finally, the users were generally positive about the amount of time they had to invest into the development process. The method does require a large amount of time and effort from the involved social scientists and ontology engineers. The stipulated guidelines allow to significantly reduce the time investment.

8.1.2 Research challenge 3: How to develop and deploy the healthcare services, which use the knowledge model, in a scalable, modular and performant manner?

To enable the development of scalable and performant healthcare services based on the developed continuous care ontology, the O'Care Platform was proposed in Chapter 3. This platform distributes the knowledge model across the various healthcare services and uses the SCB to adequately filter the huge amount of heterogeneous care data. By registering filter rules with the SCB, the various services only receive the information that they are interested in at that moment. The oNCS prototype, a localization service and a home automation application were to evaluate the scalability and applicability of the O'Care Platform.

The evaluation showed that the knowledge model can effectively be distributed across the healthcare services by employing the SCB as orchestrator. In this way, the different services only need to incorporate a small subset of the continuous care ontology to model the knowledge pertaining to their subdomain. It was demonstrated that the SCB significantly reduces the amount of data, which needs to be processed by the applications. This improves their performance and decreases overhead while maintaining an individualized approach. The delay introduced by the SCB is linear in the amount of filter rules and is negligible when 10 or less filter rules are registered. It was shown that the performance of the O'Care Platform can be increased by distributing the SCB or by distributing the filter rules across different instances of this component. A combination of both approaches can also be used. Each SCB can also employ a cache to improve its individual performance. Moreover, the platform supports the composition of complex applications from a set of smaller applications in a loosely coupled manner. The simple applications perform specific reasoning tasks in parallel and forward their conclusions to other applications by using the SCB. In this way, a truly modular development and deployment of healthcare services is achieved. Finally, as the applications can register new filter rules on the fly based on the captured information, they can readily adapt to the changing context and needs of the users.

8.1.3 Research challenge 4: How to design and develop self-adaptive healthcare services to tackle future user needs?

A self-learning and probabilistic framework is proposed in Chapter 5 as an extension of the O'Care Platform. It allows that context-aware applications adapt their behavior at run-time to the actions of the users and specific requirements of the healthare environment, in which they are deployed. It was shown how both "white-box" data mining techniques (see Chapter 5) and "black-box" machine learning techniques (see Chapter 7) can be used to realize this adaptation. The performance of the self-learning framework was evaluated with a representative use case, namely detecting SIRS as a reason for patients' call light use.

It was shown that the knowledge model and dynamic algorithms, incorporated in the context-aware applications, can efficiently be adapted by associating the new knowledge with a probability, which expresses its reliability. These probabilities ensure that the knowledge component does not become inconsistent when new knowledge is added. As the probabilities are gradually in- or decreased according to the feedback of the users, a flexible context-aware application is achieved in which new knowledge gradually comes available and old knowledge is removed based on the needs and the preferences of the users. As such, truly personalized healthcare services are achieved. Moreover, by conveying these probabilities to the users, it is ensured that they do not feel like they are no longer in control of the application as its behavior changes. Old knowledge still stays available for some time and is only gradually replaced by new knowledge, which has found wide acceptance amongst the users of the application. This increases the adoption of these context-aware services. The use of the self-learning framework also enables the stakeholders to gain insight into the way the application is used on a daily basis as they can study the learned knowledge, the associated probabilities and the fluctuations in these probabilities. Finally, it was shown that correct results were achieved for the illustrative scenario when the dataset, used as input for the self-learning framework, contains at least 1,000 instances and the amount of noise is lower than 5%. The execution time and memory usage were also negligible, i.e., below 100 ms and 10 MB. The developed self-learning framework is thus very scalable.

8.1.4 Research challenge 5: How to incorporate medical time series in the semantic model and derive useful knowledge from them?

Medical time series convey important information about the condition of a care receiver as the trend of a parameter is often more important than the absolute value of the parameter. Time series about the behavior of users also convey important knowledge. Therefore, Chapter 7 and Appendix D thoroughly investigate techniques to represent time-dependent data in the continuous care ontologies and machine learning techniques, which are able to process time series and can be used by the learning step in the self-learning framework.

It was found that the *SWRLTemporalOntology* and Echo State Networks (ESNs) can ideally be employed to reach these goals in a performant and scalable manner. A detailed comparison of ESNs to more traditional classifiers, i.e., the naive Bayes classifier and support vector machines, combined with feature extraction and selection was performed. The ESN requires significantly less processing time, needs no domain knowledge, has a comparable performance, is easy to implement, and can be configured using rules of thumb. This ensures that the ESN can easily be integrated into the self-learning framework to extract new knowledge out of time series data.

8.1.5 Detailed evaluation of the proposed solutions to the research challenges using the self-learning oNCS

To demonstrate the applicability and scalability of the combination of the continuous care ontology, the self-learning framework and the O'Care Platform, the self-learning oNCS prototype was developed and thoroughly evaluated as discussed in Chapters 4 and 6 and Appendices A and B.

Simulations and users tests showed that exploiting the profile and context information captured in the continuous care ontology results in more dynamic prioritization and nurse call assignment algorithms, which readily adapt to the current situation. These dynamic algorithms lead to a better workload distribution amongst the nurses. Moreover, they prevent that multiple nurses arrive simultaneously at the location of a patient to handle one particular call. A nurse generally also arrives quicker at the location of the call when using the oNCS instead of the current state-of-the-art nurse call systems. Finally, the simulations also showed that calls with a higher priority are generally handled faster than calls with a lower priority.

By using the O'Care Platform to filter the available healthcare data, the oNCS only receives the necessary context information. This way, a good performance is achieved, namely a suitable nurse is notified within 50.333 ms on average, which is a negligible delay. The system scales up to at least 30 patients and 20 nurses. Thus, a lot of profile information can be retained without decreasing the performance of the system. Consequently, one instance of the oNCS can for example be deployed per department in an institutionalized care setting to maintain a good performance.

It was noted by domain experts that determining the parameters of the oNCS is a difficult task. The developed self-learning framework was employed to resolve this issue. An initial assessment of the parameters is made by the domain experts when the system is deployed and the self-learning framework is used to adapt these parameters based on the context information gathered about the calls and how they are handled. It was shown that correct parameter values are achieved when the input dataset for the self-learning framework contains at least 1,050 calls. This means that one week after deployment of the oNCS in a department with 30 patients, who launch on average five calls a day, the self-learning framework would be able to correctly adjust the parameters to the behavior of the caregivers. The parameters are also learned very efficiently as the self-learning framework requires at most 100 ms execution time and 20 MB memory for a realistic dataset of 1,050 instances, irrespective of the amount of noise in this dataset.

In summary, the combination of the self-learning framework, the O'Care Platform and the continuous care ontology, which was developed in a user-driven manner by employing the designed participatory ontology engineering methodology, results in a framework, which can easily be used by application engineerings to design, develop and deploy scalable and user-centered healthcare services, which automatically adapt to the needs and the preferences of the users. Moreover, the ontology co-creation methodology can be used to design and develop new low-level domain ontologies for the various continuous care settings and application domains.

8.2 Future perspectives

This dissertation offers several contributes to the scientific research related to the design and management of ontology-based platforms for the realization of healthcare applications and services. However, research is still on-going and several open issues remain to be solved.

8.2.1 Reduce the ontology learning curve

Although ontologies have proven their merit in a number of research projects and prototypes, the transfer of these research efforts into actual products and applications is lagging behind, especially in the continuous care domain. One of the reasons is the steep learning curve of ontologies. There is a lack of guidelines, tools or rules of thumb on how ontologies and reasoning techniques can be best created for and integrated into concrete products. To resolve this, two research topics should be adressed, namely the development of design patterns for ontologies and rules and the development of an inspector tool.

The design patterns first identify ideal practices towards modeling the ontology and rules and which reasoning practices and paradigms can ideally be used for which kind of ontologies and/or rule constructs. Second, the patterns describe how the framework proposed in this dissertation can ideally be applied to a particular ontology and rules. Patterns can act as identifiers for a particular design decision or solution. This enables a clearer communication between developers.

An inspector tool can be built that takes an existing ontology and/or rules as input and detects (complex) constructs that might lead to inefficient reasoning behavior. The tool can also analyze the ontology and/or rules to identiy which reasoning paradigm can ideally be used and how. To this end, the tool can provide wizards for using the defined design patterns and can estimate the reasoning performance for the different available reasoning paradigms, which can be applied to the ontology/rules.

8.2.2 Enabling time-critical applications

The O'Care Platform was proposed in the dissertation to improve the scalability of ontology-based healthcare platforms. In this platform, the knowledge component is distributed across the different healthcare services and the SCB is used to only forward information to the services, in which they are interested at that time. As such, each healthcare application contains a smaller knowledgde model that needs to process a smaller amount of data. However, the performance of the individual knowledge components can still be a problem, especially for complex and time-critical interactive applications, e.g., a decision support tool that supports a doctor in diagnosing a patient through a question-and-answer approach. Application-specific optimalizations can be employed to achieve the needed performance constraints. This was illustrated in this research by performing the probabilistic reasoning of the oNCS at night and using these pre-processed results when a call needs to be assigned. However, it would be better to devise generic approaches towards optimizing the performance of the individual healthcare services.

To achieve more scalability, distributed reasoning techniques can be investigated, e.g., parallel reasoning or query rewriting. Active research is on-going on this topic. Another optimization is to allow incomplete reasoning, i.e., one does not wait until all the results of the reasoning are achieved, but starts working with the first results, which are gradually updated and improved as more reasoning results become available.

8.2.3 Cloud-based deployment to support homecare services

Within the O'CareCloudS project an extension of this platform to support eHomecare services is currently being researched. As the amount of elderly people and people with specific care needs increases and the number of available caregivers dwindles, a trend is emerging towards providing care at home, using smart devices and sensors. However, the homecare environments only provide limited resources to process all the data generated by the patient. As such, a cloud-based deployment of the O'Care Platform is currently being researched, in which the smart devices and sensors at the home of the patient provide their data to the O'Care Platfom deployed in the cloud and the results of the reasoning processes are send back to the patient and the appropriate associated (family) caregivers.

8.2.4 Closing the gap between folksonomies and ontologies

Folksonomies are a popular method to annotate and categorize data. Tags add metadata in the form of keywords to shared content. They are considered the electronic equivalent of Post-It notes. The totality of tags on any given information system forms the folksonomy. A folksonomy is thus a system of classification derived from the practice and method of collaboratively creating and managing tags. The main advantages of a folksonomy is that tagging is easy, hugely scalable and captures the active language of a community. Tags can be quickly created in response to new developments and changes in terminologies. As a consequence, folksonomies are entirely uncontrolled vocabularies, which is the main problem of tagging systems. Different words are used to describe the same content and folksonomies offer no mechanisms to express synonyms, homonyms or hierarchical relations. This leads to a lack of precision and recall when content about a certain topic is searched.

In constrast, an ontology formally captures the concepts in a certain domain, their attributes and their relationships and is always bound to a certain point in time and a certain point of view. Creating an ontology is laborious and requires careful consideration about how to represent a domain of interest adequately.

It can be noted that both knowledge representation paradigms have their own (dis)advantages and are quite complimentary. Some new ideas are coming up on how to combine both approaches. The main goal is to combine the popularity, convenience and flexibility of folksonomies with the semantics and high quality structures of ontologies. Ideally both approaches inspire each other within a continuous feedback loop. Ways of introducing some form of semantic control in tagging systems with ontologies are currently being explored, e.g., adding related tags based on an ontology, query expansion with semantically related tags and the ontology of tags. On the other hand, as ontology engineering is currently costly and laborious, it may profit from the huge amount of tags available. For example, a comparison of social tags and terms from a controlled vocabulary for a given domain can be performed. This helps to update existing ontologies and to evaluate the timeliness, perceivability and suitability of a knowledge representation system designed by experts.

Folksonomies can ideally be applied to integrate user-generated content into the O'Care Platform. Users, e.g., doctors, family caregivers or patients, could tag the provided data and these tags could be mapped on the ontology. The data could be integrated into the ontology by searching the concept in the formal model that is most closely related to the tag or by extending the ontology with the tag as a new concept. For the first, natural language processing can be used. For the seond, some modules of the self-learning framework could be re-used. Folksonomies are especially useful when the framework is used to support eHomecare services, as a lot of unstructured data will be provided by the patients and family caregivers.

8.2.5 Privacy, security and trust

An important acceptance criterion for rolling out the O'Care Platform is the trustworthiness of the overall system. Mechanisms need to be devised to identify the source of the provided healthcare data and identify its trustworthiness. This could be enabled by requiring that caregivers and care receivers log in with their electronic ID. As the platform can easily be extended with new applications, the trustworthiness of the data generated by these applications also needs to be assessed. Guidelines and policies can be provided, which the applications need to meet before they are allowed to communicate with the O'Care Platform. Managing the trustworthiness of the data is especially important in homecare settings, where the O'Care Platform needs to interact with a plethora of devices provided by patients.

A related issue is the security of the system and the privacy of the data inputted into and generated by the O'Care Platform. This issue is even more important in homecare environments where visitors can easily access the system of a patient. It needs to be made sure that the sensitive healthcare data of the patient is not provided to people with whom the patient does not have a personal or therapeutic relationship. Policies should be divised to determine who is able to read or adapt the information of a patient. To support the emerging trend of participatory medicine, the patient should stay in control of his/her data. This means that the patient should be able at all times to access his/her data. The implications of allowing this access, need to be studied in detail as it can have both positive, e.g., patient becomes more knowledgeable about his/her disease and takes control of his/her health, and negative implications, e.g., patient becomes anxious or tries to hide information from caregivers.

Figure 8.1: Trying to capture the fascinating and intricate healthcare domain, one vertex and one arc at a time...

A

User-driven Design of a Context-aware Application: An Ambient-intelligent Nurse Call System

F. Ongenae, P. Duysburgh, M. Verstraete, N. Sulmon, L. Bleumers, A. Jacobs, A. Ackaert, S. De Zutter, S. Verstichel, and F. De Turck

In Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), San Diego, USA, Pages 205-210, May 2012.

This chapter details the mobile application, which was used to give the domain experts a first-hand experience of the oNCS prototype during the participatory ontology engineering methodology as discussed in Chapter 2. As a result of the user feedback obtained during this evaluation, the nurse call algorithm detailed in Chapter 4 was updated. The new version of this algorithm is presented in this Appendix. The user feedback on the oNCS and the ten most important lessons learned pertaining to the development of a dynamic nurse call system are also discussed.

⋆ ⋆ ⋆

Abstract The envisioned ambient-intelligent patient room contains numerous devices to sense and adjust the environment, monitor patients and support caregivers.

Context-aware techniques are often used to combine and exploit the heterogeneous data offered by these devices to improve the provision of continuous care. However, the adoption of context-aware applications is lagging behind what could be expected, because they are not adapted to the daily work practices of the users, a lack of personalization of the services and not tackling problems such as the need of the users for control. To mediate this, an interdisciplinary methodology was investigated and designed in this research to involve the users in each step of the development cycle of the context-aware application. The methodology was used to develop an ambient-intelligent nurse call system, which uses gathered context data to find the most appropriate caregivers to handle a call of a patient and generate new calls based on sensor data. Moreover, a smartphone application was developed for the caregivers to receive and assess calls. The lessons learned during the user-driven development of this system are highlighted.

A.1 Introduction

The envisioned ambient-intelligent care room [1] comprises plenty of sensors to sense the needs and preferences of the staff and patients and devices that work together to adapt the environment to support them in carrying out their daily activities. To realize this vision, context-aware techniques are often used to combine and exploit the heterogeneous data offered by all this technology to improve the provision of continuous care [2]. E.g., if the system is able to determine the caregiver's task and the patient's condition, it can automatically adapt the environment to their needs, e.g., adjust the light level or show relevant information about the task.

However, the adoption of context-aware services is lagging behind what could be expected. Whereas the healthcare industry is quick to exploit the latest medical technology, they are reluctant adopters of modern health information systems [3]. Half of all computer-based information systems fail due to user resistance and staff interference [4]. The main complaint made against mobile, context-aware systems is that users had to significantly alter workflow patterns to accommodate the system [5]. This is due to inadequate techniques for personalization of the services, a lack of focus on the soft aspects of interaction, e.g., automated and personalized alerts, and the lack of tackling problems such as the need of the users for control [6]. To ensure that technology and environment blend into each other, the users should be involved in each step of the development cycle of the applications [7].

Therefore, an interdisciplinary methodology was designed to develop a prototype context-aware application. Social scientists, engineers and users, e.g., doctors, caregivers and healthcare industry professionals, were involved in every step of the development process. The research started from the needs and daily work practices of the stakeholder to determine the ideal prototype application to develop. It was found that a nurse call system is an important way to coordinate work, communicate and provide continuous care.

Traditional nurse call systems are static as calls are made by buttons fixed to a wall and the nurse call algorithm consists of predefined links between rooms and caregivers' beepers [8]. They do not take into account the current situation to assist the user in making calls, assign a nurse to a call or detect hazardous situations for which a call should be made. Moreover, the beepers give the caregivers limited context information about the call.

In this research, the user-driven approach was used to develop a dynamic, ambient-intelligent nurse call system. It integrates the heterogeneous data collected by the devices, e.g., location data, medical parameters and domotics data. The system uses this information to find the most appropriate caregiver to handle the call of a patient and even to generate calls based on the context information, e.g., when a patient spikes a fever. Moreover, a smartphone application was developed, which is used by the caregivers to receive calls, assess & redirect them, contact the patient, etc. The users were involved in each step of the development process of this ambient-intelligent system to determine the prevalent context information that should be taken into account, the algorithms which should be used to generate, assign and prioritize nurse calls and the requirements and user interface of the mobile application.

The remainder of this paper is structured as follows. Section A.2 details the ambient-intelligent nurse call system and developed mobile application. Section A.3 discusses the user-driven methodology which was used to develop this system. The lessons learned from designing the context-aware application with this methodology are discussed in Section A.4. Finally, Section A.5 highlights the conclusions and future work.

A.2 The ambient-intelligent nurse call system

A.2.1 General architecture

The architecture of the ambient-intelligent nurse call system, which was developed using the user-driven methodology described in Section A.3, is shown in Figure A.1. Each patient and caregiver has a badge to locate this person. Each badge also has a call button allowing patients and staff to walk around freely and still make (assistance) calls. The ambient-intelligent care environment contains numerous devices & sensors that sense the context and collect information about the environment. A desktop provides the head nurse with a user-friendly interface to input and visualize information about the department, e.g., the number of patients and available caregivers and their characteristics and roles. Each staff member is notified of calls assigned to him/her by a smartphone application, which is discussed further in Section A.2.2.

The *Context-aware Platform* [9, 10], depicted at the top of Figure A.1, handles the communication to and from all the devices and sensors. The *Context Interpreter* uses an ontology [11] to interpret the provided heterogeneous data. An ontology formally describes the concepts in a domain, their relationships and attributes. The used ontology models all the necessary context information about the continuous care domain, e.g., the profiles of the staff & patients, the possible tasks & calls and knowledge about the devices and sensors. This ontology was developed using a participatory ontology engineering methodology, as discussed in Bleumers, et al. [12].

When new data is inserted in the ontology, the *Context Interpreter* uses reasoners [13] and rules to infer new knowledge out of this information. For example, when a new call is inserted, the *Context Interpreter* assigns the most appropriate staff member to this call based on the available context data using the algorithm that is discussed in Section A.2.2.

The *Context Providers* are responsible for translating the information, delivered by the various devices and the database, to data that can be inserted in the ontology. The *Query Services* do the exact opposite, they transform the data and conclusions inferred by the *Context Interpreter* to information that can be processed by the various devices. This can result in changed status of a device, e.g., dimming a light, or in a message that alerts a staff member, e.g, about an assigned nurse call.

A.2.2 Mobile nurse call application and nurse call algorithm

The ambient-intelligent nurse call system differentiates between 3 types of calls. Normal calls are initiated by patients pushing a button. Caregivers can launch assistance calls to ask for help by pushing a call button or the orange hexagon on the mobile application, as shown at the upper right of Figure A.2a. Finally, context calls are generated by the nurse call system as a consequence of measured sensor values, e.g., a temperature sensor indicates that a patient is spiking a fever.

When the ambient-intelligent system receives or generates a call, the rule-based algorithm finds the most appropriate caregivers to handle it. It first determines the patient for whom the call is made. Next, the algorithm finds all the staff members who have a high degree of trust relationship with this patient, e.g., a therapeutic or personal relationship. If no such staff members are found, this step is ignored. Out of these filtered staff members, caregivers are preferred who are close to the patient and not busy with a high priority task. This algorithm allows rapidly finding caregivers to initially assess the call.

When the staff receive the assigned call on their smartphones, it vibrates to avoid noise overload. As shown in Figure A.2a, the associated patient, the number

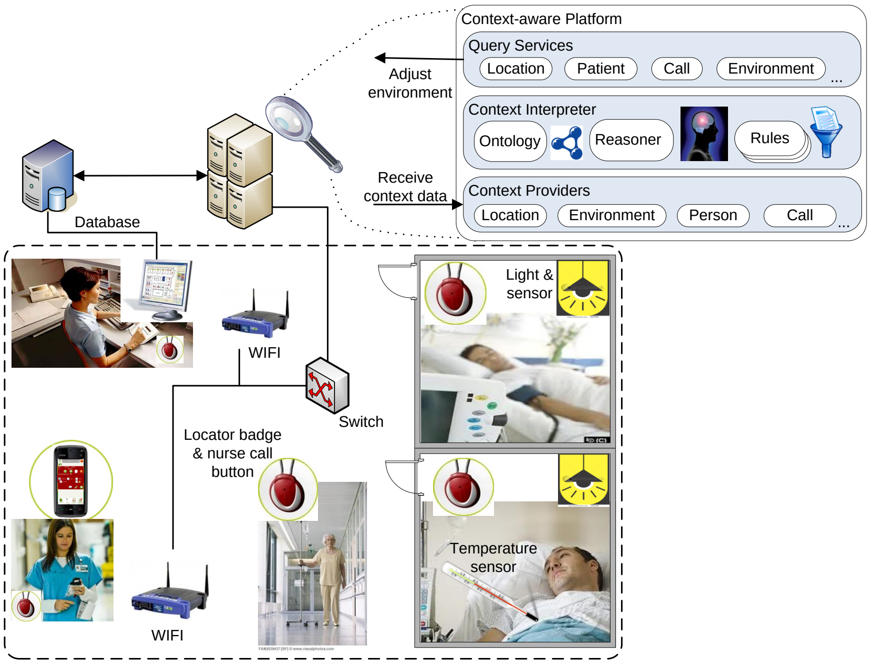


Figure A.1: General architecture of the ambient-intelligent nurse call system.

of times he/she has pushed the button and the location, type and timestamp of the call are visualized. For a context call, the sensor and values that caused the call to be generated are also shown. The caregiver can decide to go to the patient and handle the call, but he/she can also contact the patient to assess the reason and importance of the call by clicking the green telephone icon. A telephone call is made to the handheld device of the patient to preserve privacy. However, if the patient does not pick up after three rings, the call is established through the intercom in the room terminal. After contacting the patient, the caregiver can triage the call, as shown in Figure A.2b, by indicating whether the reason is a caring task, medical task or hotel funtion, e.g., a glass of water. As depicted in Figures A.2c, A.2e and A.2d, the color of the call changes to reflect its reason. After this assessment, the caregiver can indicate on the smarthpone that he/she is going to handle the call. The call then disappears from the smartphones from the other assigned caregivers. As visualized in Figure A.2a, the person who accepted the call can still see it by clicking the button "accepted calls" in the lower right corner. It indicates the number of accepted calls in a red circle. The caregiver can also decide to add information to the call, e.g., by jotting down a note (pencil button) or changing the reason. Finally, the caregiver can also finish the call remotely by clicking the white paper button. The context of the call and the information provided by the caregiver are automatically transferred to the care registration file of the patient, which can be checked by clicking the lower left button.

However, the caregiver can also decide to redirect the call, e.g., because he/she does not have the required competencies, by pressing the green right arrow, shown in Figure A.2a. Figures A.2c, A.2e and A.2d show the redirection screen of a call with a care, hotel function and medical reason respectively. For the latter category it can also be specified that a doctor is needed. The caregiver can change the reason, indicate that the call is urgent or add a note. When the call is redirected, the nurse call system uses a more complex algorithm to find staff members to handle the call, which takes into account the context information provided by the first caregiver. As the reason of the call is known the algorithm first filters the staff members with the appropriate competencies to handle the call. The algorithm prefers staff members who have this competence as part of their current role, but it also considers caregivers who have these competencies through secondary roles, separately acquired competencies or experience. If no staff member is found with the required competencies, this step is skipped and the previously detailed nurse call algorithm is used. As it is most important for urgent calls that a caregiver is quickly able to handle the patient, the algorithm does not take the trust relationship into account for these calls. More weight is thus given to the distance and current task parameters.

The newly assigned staff members receive the redirected call on their smartphone as shown in Figure A.2f. If the call has priority urgent, the smartphone rings instead of vibrating. The caregiver can contact the person who redirected the call and access all the information that was previously provided. This staff member can also decide to immediately handle the call, to redirect, accept or finish the call or to contact the patient.

To illustrate the integration of the nurse call system with the devices in the environment, it does not only generate calls based on gathered sensor data, but also adjusts the light level in the room based on the reason of the call and the presence of caregivers and unlocks the supply closet when a person with the appropriate competencies logs in on the room terminal.

The next Section describes how the user-driven desing process helped to shape the described context-aware application.

A.3 User-driven design

A.3.1 Observations: define goals and scope of the prototype

The user-driven design started with a user and task analysis. By observing and interviewing the target users in their environment, their needs and wishes about their daily tasks were determined. Two types of care settings were observed: a

(a) New call (b) Contact patient (c) Triage: care

(d) Triage: hotel (e) Triage: medical (f) Redirected call

Figure A.2: Screenshots of the user interface of the mobile nurse call application

hospital and a residential care setting for people with a cognitive and/or physical impairment. The observations focussed on the communication between caregivers and with their patients.

It was observed how stakeholders use their nurse call systems. Within the team, there are routines about who handles which patients and people communicate when they are temporarily unavailable. The caregiver needs to go to the room of the caller to determine the urgency and reason of the call and whether they need additional care prodcuts to handle it. This leads to a lot of extra miles for the caregivers and a need to interrupt their current tasks to assess the call. In care settings there are many sound signals, e.g., beepers, phones ringing and monitoring equipment. Participants noted they cope with a sound signal overload and became immune to it. Caregivers working the night shifts mentioned that sleeping patients woke up bacause of their beepers. Moreover, staff members do not always take their beeper with them as they find the beeping annoying when they are helping someone and cannot leave anyway. From a patient view, a lack of feedback after making a call was observed. Patients were left with questions, e.g., “Did they hear me?” and “How long will I need to wait?”. Finally, a high demand of care registration at the point and time of care was observed. This is now done after the shift.

In the hospital setting, the nurse call system had a room to room intercom feature. This allowed a nurse to contact patients before coming to their room. Even though this feature could provide additional information and give feedback to the patient, it was not used. This was due to privacy reasons as other patients in the room could follow the conversation. Also, patients mentioned that it was awkward to hear a voice in the room, without being able to determine where it came from.

Depending on these users’ needs and abilities, the following requirements were derived for the ambient-intelligent nurse call system. The nurse call algorithm should take into account context, e.g., the walking distance and the availability, role & competences of the caregiver. To allow mobile care registration and requests, each patient should have a mobile nurse call button and each staff member should have a smartphone with a care application. Moreover, detailed information about the call should be visualized on the smartphone, e.g., who and where is the patient and the reason, urgency and timestamp of the call. This demands a way to localize the patients and caregivers at all times. The application should assist the caregiver with registering information about the call on the fly. The smartphone should allow the caregiver to contact the patient from anywhere in the environment to provide feedback to the patient. To preserve privacy and confuse the patient less, contact should be established through a personal device of the patient, e.g., a handheld device or the wheelchair. However, if the patient cannot be reached, the intercom should be tried. To decrease the noise overload, the smartphone should vibrate instead of ring when a call is received that is not urgent. Finally, when the system is installed, continuous training should be given about all the features to increase acceptance.

A.3.2 White book & sunny-day scenario

In order to keep an overview of the requirements and objectives of the novel ambient-intelligent nurse call system, a *white book* was created as central coordination instrument between the software engineers, user researchers and stakeholders. The construction of the white book was started after the first observations, but the document continued to grow and adapt during the whole development cycle of the system.

The white book starts with the description of various personas. Personas highlight the representative user archetypes of a system, the activities they wish to perform, why they wish to use the system and how the system fits into the context of their life. Their main advantage is that they allow feeling empathy for the user group, as they put a human face to a list of requirements. As such, they explain the origins of the requirements and why certain design decision are made. In total 13 personas were created. The persona Erik lives at a care residence, has Duchenne disease and is dependent on a wheelchair. Personas were also created for Erik's parents and brother, staff at the care residence and associated hospital.

Second, a sunny-day scenario is described. A scenario is a story that describes the hypothetical use of a system to help develop a detailed and shared understanding of the context and activities of the users. The scenario consists of a number of scenes in which the actions of the personas are described such that the functionalities of the novel system become clear. The scenario is sunny-day because it is unconstrained by current technological possibilities. The scenario starts with a description of how the nurse call system would be installed and caregivers would be trained. Next, a night in the life of Erik is described in which he makes calls, the caregivers use the system to ask for assistance and the novel nurse call system is used to ideally handle these situations. Next, Erik spikes a fever and a context call is generated and assigned. Erik is transferred to the hospital, where he also makes nurse calls to illustrate the use of the system in this setting.

Third, the ICT equipment needed to realize this scenario is described, e.g., the locator badges, temperature sensors, smartphones and call buttons. Finally, the white book describes the translation of the sunny-day scenario to a prototype implementation that can be technically realized. The architecture of the nurse call system and the user interfaces of the designed mobile application, as shown in Section A.2, are detailed.

The white book was evaluated and adapted together with the users at multiple occasions. The scenario was also used as a basis for several workshops. The evolution of the scenario was detailed in the white book with clear links to workshops and user interactions that triggered the changes and insights.

A.3.3 Decision-tree workshops

The observations and the first version of the white book allowed to capture the scope, requirements and needed context information for the ambient-intelligent nurse call system. However, it was difficult to distill the decision process that caregivers propose or find ideal to prioritize and assign nurse calls. To resolve this, decision-tree workshops were organized.

At the start of the workshop, the participants described a complex situation involving nurse calls. Next, participants were asked to suppose they were an intelligent system that had a complete overview of the current situation. This system takes patients' nurse calls as input and is tasked with prioritizing and assigning the most appropriate caregivers to the call. The real life situations described by the participants were used to start the discussions by visualizing them, e.g. location of the patient, on a blue print of the work environment of the participants. To gather more context and make an informed decision, the participants asked questions. Instead of answering the question, discussions were first held about the importance of the requested info and possible answers the participants envisioned. This way the user researchers could tap into the reasoning made by the participants. The technical engineer visualized these questions on paper in the form of a decision tree. The order of the information in the tree reflects its importance, while the different nodes represent the parameters that should be taken into account to reach the ideal assignment.

It was determined that the assignment of caregivers to calls should depend on, in order of importance, the reason of the call, the competencies & roles of the staff, the priority of the call, the trust relationship and distance between the caregivers and patient and the current tasks of the staff. Consequently, several changes to the white book were made. Taking into account the roles, competencies and trust relationship was deemed much more important as the researchers perceived during the observations, while distance was deemed much less important by the participants. It was also assumed that a whole plethora of priority levels should be assigned to calls as this is usually the case in traditional nurse call systems. However, the participants claimed they only discerned between 3 levels, namely normal, urgent and very urgent. The latter category is preserved for life-threatening situations. Finally, participants also desired to redirect calls, easily get in touch with the staff member who redirected it and add information to a call, e.g., the reason which was discerned by contacting the patient.

Initially, the sunny-day scenario described the system as being able to determine the reason for a call based on the gathered context data. However, the participants perceived this as unrealistic, as such insight is achieved by years of experience and a deep understanding of the patient and situational context. They feared an unstable, incorrect and controlling system. However, the participants did conclude that for assigning staff members to calls, 3 main reasons for making a call need to be discerned, namely hotel, caring and medical reasons.

Consequently, it was decided to not let the nurse call system determine the reason and the priority of the call. To replace this, the possibility to contact the patient, triage the call, assess its priority and redirect it was added to the mobile application. At this point the nurse call algorithm was split up in a simple algorithm to quickly assign calls to initial staff members and a second, more intelligent algorithm which is used after a call is redirected, as detailed in Section A.2.2. An algorithm was preferred above letting caregivers choose specific staff members to whom the call should be redirected. This is easier for inexperienced staff, frees caregivers from remembering who is currently available, increases the workload distribution of the calls and allows to take into account other context parameters when assigning calls. Finally, to better illustrate the benefits of using the ambient-intelligent nurse call system, the following features were added: generating context calls, adjusting the light level in the room based on the kind of call and presence of staff and unlocking the supply closet when a person with the appropriate competencies logs in.

A.3.4 Concept evaluation workshops

The purpose of these workshops was to do some preliminary testing of the conclusions and changes that were made with regard to the white book and the ambient-intelligent nurse call system after the decision tree workshops before implementing them. Two types of workshops were organized. In the first workshop, the functionalities of the system were evaluated by participants with various qualifications, e.g., nurses, doctors, domains experts and designers. To illustrate the novel system, a movie was made of a specific part of the white book scenario, where most innovative functionalities were introduced. This movie was first shown in its entirety and then paused when elements were introduced that researchers wanted to discuss with the participants in smaller groups, such as the triage and the use of mobile devices. The second workshop consisted of individual usability tests of the preliminary interface design of the mobile application. The participants were presented paper prototypes of the interface. After a short introduction, the participants were asked to perform a task on the interface, without receiving instructions about the functions of the buttons. The participants were asked to talk out loud and explain what they did and thought that the symbols represented.

Both workshops resulted in useful feedback. The idea of call triage generated enthusiasm amongst the participants. The use of mobile devices caused some concern with regard to hygiene. There was some discussion whether the devices should vibrate or make a sound when a call came in, or if a mixed solution could be found. Also, there was a lot of discussion on how trust relationships should be integrated in the system. In addition, some participants found it difficult to redirect calls only to a certain “profile” rather than to a specific person. The usability tests led to some minor adjustments in the design of the user interface of the application, e.g., changing and moving buttons, adding feedback messages to indicate that an action was successful and translating the application into dutch.

A.3.5 User evaluation: embodied system use

To achieve a deeper reflection on the novel ambient-intelligent nurse call system by the users, a prototype was implemented, as detailed in Section A.2, in the Patient Room of the Future (PRoF). PRoF is an intelligent patient room and adjacent hallway, realized in Belgium, aimed to make a patient feel more like home. For the prototype, RF tags and receivers were integrated to track the locations of the patients and staff. Temperature sensors were also available to monitor the temperature of the patients. The developed ambient-intelligent nurse call system was installed in PRoF and integrated with the available light control system, RF tags and sensors. Smartphones running the designed mobile application were also provided. This prototype allowed users to experience a fully immersed, more profound, contextual experience of the system in a lifelike context. After an elaborate introduction of the system, the participants were given context and persona cards. The context cards included instructions, which participants were asked to play out and resembled their professional activities. The persona cards identified the role they played, e.g., patient or nurse. In between and after the scenes, the participants discussed the system and mobile application with the researchers. During the first sessions, technical issues sometimes interfered with the role-play. These were solved and were no issue in the other sessions.

The evaluation resulted in a lot of recommendations that will be solved in future work. The way the trust relationship was integrated in the system was too rigid and decisive. Although the participants liked the idea of triage and redirecting the calls, some issues were noted. After redirecting a call, the caregivers sometimes felt the need to contact the caregiver who had finally handled the call to know how the problem was solved. Moreover, after a staff member had contacted the person who redirected a call, it was sometimes requested to be able to send the call back to this person. Also in this workshop, some participants had difficulty thinking of their colleagues in terms of their qualifications and felt the need to redirect calls to specific colleagues. Also, it quickly became clear that the smartphones should not only vibrate but also need to give an audio signal. Although some of these issues can be explained by the participants’ current work practices, it also makes clear that extra attention should be paid to the adaptation of new work practices when the system is implemented in a real-life environment, since this might form a threshold for adoption.

A.4 Discussion

This paper illustrated how an interdisciplinary research team made an ambient-intelligent nurse call system in close collaboration with the users and stakeholders. An important lesson learned is that an intelligent system does not have to determine and solve everything as the users of the system are sometimes better suited to make decisions, e.g., triaging the calls. It became clear that a context-aware system in care should support caregivers and facilitate for instance data integration, but should also allow caregivers to overrule the system and have control over their work flow and environment.

Observations proved to be insufficient as user input. Only by repeatedly involving users throughout the design process, the researchers sufficiently nuanced their understanding of the users and their context to make a system that supports the users' daily work processes, without making them feel like they lost control. This is not to say that the described user involvement could not be improved. Although the final tests took place in PRoF, which was very close to reality, it was felt that a real-life setting could generate further insights. It will be investigated how a mobile set-up of the system can easily be tested in a real-life work setting. However, the varying available technology and networks make this a challenging endeavour. During the final tests, some technical issues popped up, which threatened to reduce the user tests to technical tests. Although these issues were quickly solved, it was sometimes hard to distinguish the participants' feedback on the system from feedback related to technical system failure.

Below the ten most important findings from the user-driven design process are summarized.

1) The novel nurse call system requires the users to think of their colleagues in terms of their qualifications and let the system redirect the call. The caregivers had a tendency of thinking of a specific colleague best suited for the job.

2) During workshops in the residential home, the trust relationship was regarded as a decisive element for assigning calls. However, it proved to be hard to translate this to an algorithm without creating too much side effects.

3) Early on it was decided that the smartphones should alert a call by vibrating to avoid noise overload. However, nearly all users in the final tests still requested a sound signal.

4) When redirecting a call, the user tests revealed that most participants like to know how the call was handled in the end as this gives them a sense of control and overview. .

5) The users did not want the system to *dehumanize* their interactions with patients and colleagues. They liked that they could contact the patient after receiving a call.

6) Similarly, it was decided that the triage should not be done by the system, but by the caregiver talking to the patient.

7) The notion of distance had to be repeatedly discussed and reinterpreted. Based on the observations, the system was designed such that the caregivers would have to walk smaller distances. However, during the workshops, it became clear that other elements were considered equally important to dermine who should handle a call. As such, 'distance' became one of the parameters taken into account, rather than a decisive one.
8) While assigning calls more directly was seen as a big advantage of the system, participants noted that this implied losing an overview of what was going on at the department. Although this could partly be resolved through informal contacts, other general indications of activity will be needed.
9) Participants worried about implementation every time the system was presented as it require a new mindset and an alternative way of perceiving their colleagues. Moreover, care institutions have a lot of interns and a frequent change of personnel. This obstructs the adoption of new technology and functionalities. Given the additional fact that this system differs substantially from other systems and the current work practices, considerable attention should be paid to change management when implementing it.
10) An important challenge during all workshops was to explain how the system worked before any feedback could be gathered. In general, there was some discussion to what extent the users should understand the full complexity of the system when starting to use it. This is important to consider when looking at change management and implementation.

A.5 Conclusion

This paper used an interdisciplinary user-driven methodology to design and develop an ambient-intelligent nurse call system and smartphone application. This way, the system is tuned towards the daily work processes, wishes and needs of the users. Moreover, the user-driven approach *humanizes* the system, increases its acceptance and makes the users feel in control. Future work will focus on incorporating the recommendations of the embodied user tests and investigating a way to create a general overview of the current situation in the department. Methods will also be investigated to easily test a mobile set-up of the system in a real-life setting.

Acknowledgment

Part of this research was performed in the PRoF1.0 demo room and supported by the ACCIO Project co-funded by the IWT, IBBT and following partners: Televic NV, Boone NV, Dominiek Savio Instituut and In-Ham. F. Ongenae thanks the IWT for her Ph.D. grant.

References

[1] Y. Punie. *The future of ambient intelligence in Europe: the need for more everyday life*. Communications & Strategies, (57):141–165, 2005.

[2] J.-C. Burgelman and Y. Punie. *Close encounters of a different kind: ambient intelligence in Europe*. True Vision: The Emergence of Ambient Intelligence, pages 19–35, 2006.

[3] T. Chin. *Technology Valued, but Implementing it into Practice is Slow*. American Medical News, 2004. http://www.ama-assn.org/amednews/2004/01/19/bisb0119.htm.

[4] J. Anderston and C. Aydin. *Evaluating the Impact of Health Care Information Systems*. International Journal Technology Assessment in Health Care, 13(2):380–393, 1997.

[5] J. H. Jahnke, Y. Bychkov, D. Dahlem, and L. Kawasme. *Context-aware information services for health care*. In Proc. of the Workshop on Modeling and Retrieval of Context, pages 73–84, 2004.

[6] J. Criel and L. Claeys. *A transdisciplinary study design on context-aware applications and environments. A critical view on user participation within calm computing*. Observatorio, 2(2):57–77, 2008.

[7] N. Bricon-Souf and C. R. Newman. *Context awareness in health care: A review*. International Journal of Medical Informatics, 76(1):2–12, 2007.

[8] E. T. Miller, C. Deets, and R. Miller. *Nurse call and the work environment: lessons learned*. Journal of Nursing Care Quality, 15(3):7–15, 2000.

[9] F. Ongenae, A. Ackaert, F. D. Turck, A. Jacobs, A. Veys, M. V. Gils, and P. Verhoeve. *User-driven design of an ontology-based ambient-aware continuous care platform*. In Proc. of the 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Munich, Germany, 22-25 March 2010. Piscataway, USA: IEEE.

[10] F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck. *An ontology-based nurse call management system (oNCS) with probabilistic priority assessment*. BMC Health Services Research, 11:28, 2011.

[11] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, M. V. Gils, A. Jacobs, S. De Zutter, P. Verhoeve, A. Ackaert, and F. De Turck. *Participatory design of a continuous care ontology: towards a user-driven ontology engineering*

methodology. In Proc. of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, 2011.

[12] L. Bleumers, N. Sulmon, F. Ongenae, A. Jacobs, M. Verstraete, M. Van Gils, A. Ackaert, and S. De Zutter. *Towards ontology co-creation in institutionalized care settings*. In Proc. of the 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pages 559–562, Dublin, Ireland, 2011.

[13] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL Reasoner*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 5(2):51–53, 2007. http://pellet.owldl.com/.

B

Probabilistic Priority Assessment of Nurse Calls

F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck.

Revisions Submitted to Medical Decision Making, March 2013.

This appendix elaborates on the oNCS, which was presented in Chapter 4. It focuses more on a detailed description of the probabilistic algorithms used to determine the probability that a call has a specific priority and the threshold algorithm used to determine the most suitable priority for a call based on these probabilities. This appendix also discusses the different existing approaches for respresenting and reasoning with probabilistic information in ontologies, motivates the choice for Pronto and evaluates its performance. Moreover, data was gathered about two additional departments at Ghent University Hospital. Their simulation results are discussed and compared to each other and to the results of the first department of which the results were already discussed in Chapter 4.

⋆ ⋆ ⋆

Abstract The current nurse call systems are very static. The call buttons are fixed to the wall. Additionally, these systems do not take into account various factors specific to a situation. We have developed a software platform, the *oNCS*, which supports the transition to mobile and wireless nurse call buttons and employs an

intelligent nurse call algorithm. This algorithm dynamically adapts to the situation at hand by taking the profile information of staff and patients into account by employing an ontology. This paper describes a probabilistic extension of this *oNCS* that supports a more sophisticated nurse call algorithm by dynamically assigning priorities to calls based on the risk factors of the patient and the kind of call.

The probabilistic *oNCS* is evaluated through a prototype implementation and simulations, based on a detailed dataset obtained from 3 nursing departments of Ghent University Hospital. The arrival times of nurses at the location of a call, the workload distribution of calls amongst nurses and the assignment of priorities to calls are compared for the *oNCS* and the current nurse call system. Additionally, the performance of the system and the parameters of the priority assignment algorithm are explored.

The execution time of the nurse call algorithm is on average 50.333 ms. Moreover, the probabilistic *oNCS* significantly improves the assignment of nurses to calls. Calls generally have a nurse present faster, the workload-distribution amongst the nurses improves and the priorities and kinds of calls are taken into account.

B.1 Introduction

In recent years the complexity of continuous care has been increasing due to the increase of the care unit size, specialized care and combined care paths. The lack of nurse staffing also requires a more efficient use of resources. To deal with these issues, information technology (IT) is often used. IT has already proven its merit in other healthcare fields [1–3].

Observations and contextual inquiries at a representative hospital setting at the start of this research [4, 5] revealed that ICT could greatly contribute to continuous care by performing 1) information integration & data provisioning at the point of care and 2) by supporting communication between both the staff and the staff and the patients. The first can be summarized as providing the right information, at the right time, at the right place for the right person. This requires an increased need for mobile services [6] to support data input, e.g., care registration, and request data, which should be integrated, prioritized and filtered based on contextual information.

Regarding the second point, it was found that a nurse call system is an important way to coordinate work, communicate and provide continuous care. When patients feel unwell they push a button. The nurses then receive a message in a beeper with the room number. This brings the question: which nurse goes to the room, the closest one, the one on call, etc.? The caregivers use the nurse call system to be alerted of patients' needs, communicate with them through intercoms and request help from other staff members.

Traditional nurse call systems are however very static as calls are made by buttons fixed to a wall and the nurse call algorithm consists of predefined links between rooms and caregivers' beepers. Herewith two important assumptions are made: the patient must still be in the room when the assigned nurse arrives and it must be the patient who lies in the room that made the call. The current systems thus do not take into account the specific situation and context, such as the risk factors of a patient or the locations of the staff, to assign a nurse to a call. The beepers give the nurses limited context information about the call. They need to go to the room of the caller to determine the urgency and reason of the call and whether they need additional care products to handle it. This leads to a lot of extra miles for the caregivers and a need to interrupt their current tasks to assess the call. Nurses are also not aware of each other's tasks and whether a staff member already has the intention to handle the call. This causes that multiple nurses arrive at a room to handle a call and thus that their tasks were unnecessarily interrupted. Moreover, it is dangerous for patients to become unwell inside a hallway, staircase or outside as patients cannot call a nurse in these areas. This leads to patients being confined to their room to ensure their safety.

A trend is already emerging towards nurse call systems, which are equipped with a mobile button for each patient so that they can walk around freely [7, 8]. Novel nurse calls systems also equip each room with a terminal screen. However, this screen is currently not used to display relevant context data.

It is thus clear that continuous care could greatly benefit from the incorporation of a context-aware nurse call system which uses integrated context information about the staff and patients, such as their locations and qualifications, to assign the appropriate nurse to the patient. This way the communication and workflows could be optimized in a dynamic way. This is necessary as calls are unforeseen tasks with a wide variety of reasons and priority, which makes them difficult to assign and schedule in advance. Moreover, the relevant context information pertaining to the current call could be visualized to the assigned caregiver and data input about the call could be supported at the point of care.

To realize this vision, we have designed the ontology-based Nurse Call System (*oNCS*). This platform allows that patients walk around freely with portable, wireless nurse call buttons. Additionally, this platform efficiently manages the profiles of the staff members and the patients by encoding this information into an ontology [9]. An ontology formally models all the concepts and their relationships and properties within a domain. A new nurse call algorithm was developed, which dynamically adapts to the situation at hand by taking this profile information into account, to find the best staff member to handle a specific call. This makes the system more adaptable to the needs and preferences of the patients and staff members. A description of this platform can be found in Ongenae, et al. [10].

The goal of this paper is two-fold. First, an extension of the *oNCS* is described in Section B.2, which supports a more sophisticated nurse call algorithm by dynamically assigning priorities to calls based on the risk factors of patients and the kind of call. As patients with a particular profile can still make calls of varying priority, this information is modeled probabilistically in the ontology. By using probabilistic reasoning and threshold algorithms, all these probabilistic values are combined in an intelligent manner to determine the most suitable priority for a call. These priorities are then taken into account when a suitable nurse is searched to handle a call. Second, an extensive simulation with realistic data about 3 departments of Ghent University Hospital [11] is presented in Section B.3 that demonstrates the advantages and performance of the system and explores the sensitivity of the threshold algorithm. A critical discussion of the platform and its expected benefits is presented in Section B.4. Finally, Section B.5 highlights the main conclusions. This paper focuses on the probabilistic modeling of the data and accompanying probabilistic reasoning algorithms. An overview of the design of the complete system can be found in Ongenae, et al. [12].

B.2 Methods

B.2.1 Profile Management

In order to achieve a nurse call algorithm that adapts to the situation at hand, an ontology is used to efficiently manage the available context information. From field trials and experiments it was derived which context information is relevant. The resulting ontology is visualized Figure B.1. First, the different staff members and their properties were modeled such as their location, characteristics, the departments they work on, their current status (free or busy) and their current task. Second, information about the patients was modeled such as their location, risk factors, department and the characteristics they prefer in staff members. Finally, the various calls and tasks were modeled. Three kinds of calls can be launched by patients. A normal call is made for medical problems and a service call is made for a “caring” task, e.g., asking for a glass water. Sanitary calls originate from a sanitary space, e.g., bathroom. All the other calls, namely urgency, medical, technical and (sanitary) assistance calls, are launched by nurses. For each call it is also indicated which kind of staff member may handle it. Each task and call has an associated priority. A more detailed description of this ontology can be found in Ongenae, et al. [12].

This ontology was extended with profile information to probabilistically determine the priority of a call. First, the risk factors were added to the ontology. A complete list of risk factors could be constructed based on a thorough study of the risk factors of patients and the reasons for the calls that they make. Such studies have not been conducted to the knowledge of the authors. To highlight the possibilities of the system, a list of risk factors was assembled by experts from both the

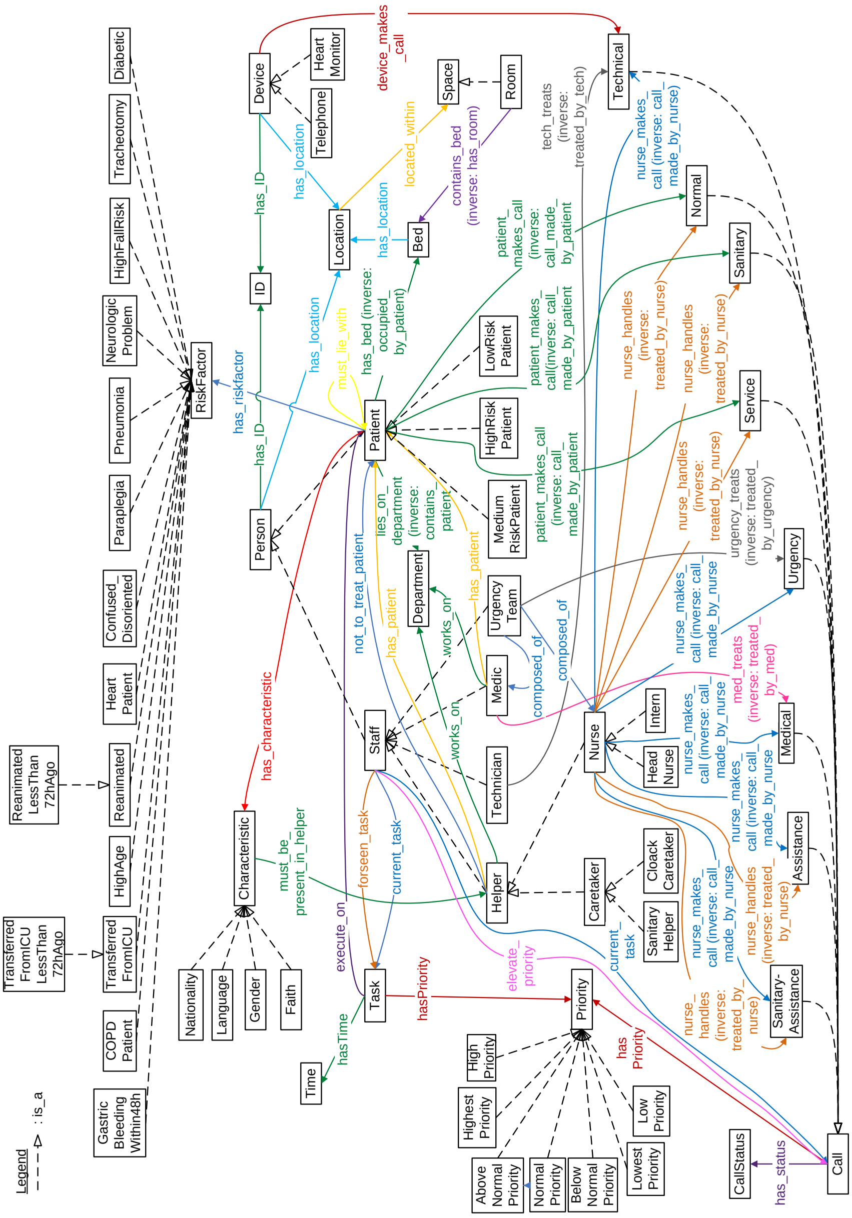


Figure B.1: The ontology modeling 1) the profile information of nurses and patients and 2) the types of calls that can be launched and who can handle them. The squares symbolize concepts. The black, dashed arrows represent subclass relationships (is a). The colored arrows indicate relationships between concepts.

Patient has risk factor:	High Risk	Medium Risk	Low Risk
Elderly (high age)	[0,0.2]	[0.5,1]	[0,0.3]
Diabetes	[0.5,1]	[0,0.3]	[0,0.2]
Heart patient	[0.5,1]	[0,0.4]	[0,0.1]
High fall risk	[0.7,1]	[0,0.2]	[0,0.1]
Neurological problem	[0.5,1]	[0,0.3]	[0,0.2]
Tracheotomy	[0.8,1]	[0,0.2]	[1,0]
COPD patient	[0.6,1]	[0,0.3]	[0,0.1]
Paraplegic	[0.6,1]	[0,0.3]	[0,0.1]
Pneumonia	[0.5,1]	[0,0.4]	[0,0.1]
Gastric bleeding within 48 hours	[0.8,1]	[0,0.2]	[1,0]
Transferred from the ICU	[0.6,1]	[0,0.4]	[1,0]
Transferred from the ICU within 72 hours	[0.7,1]	[0,0.3]	[1,0]
Reanimated	[0.6,1]	[0,0.4]	[1,0]
Reanimated within 72 hours	[0.8,1]	[0,0.2]	[1,0]
Confused or disoriented	[0.6,1]	[0,0.3]	[0,0.1]

Table B.1: The probabilistic assignment of patients to risk groups based on their risk factors

medical and nurse call domain.

When a patient exhibits a risk factor, he is assigned a probability of belonging to a risk group, namely High, Medium and Low Risk Patients. The assigned probability is also influenced by the department where the patient resides. A preliminary set of probabilities, as determined by domain experts, is shown in Table B.1. The probability interval [1,0] expresses that a patient with this risk factor never belongs to this risk group. Of course patients can have several risk factors, in this case the system reasons over the different probabilities to determine the probability that a patient belongs to a particular risk group, as explained in Section B.2.2.1.

The probabilistic assignment of patients to risk groups is used to determine the priority of the calls made by or for these patients. There are seven classes of priorities: Highest, High, Above Normal, Normal, Below Normal, Low and Lowest priority. The priority of a call is also based on its kind, e.g,. normal or sanitary. So when a patient from a particular risk group, makes a particular kind of call, this call is assigned a probability of having a specific priority. These probabilities were determined by domain experts at Televic NV [13] as shown in Table B.2. Combinations with probability interval [1,0] are depicted with an empty place in the table to prevent cluttering. Strict probabilities were used here, but they can easily be translated to intervals by using equal upper and lower limits. Note that urgency, medical and technical calls generally get the highest, low and lowest priority respectively. Thus, the priority of these calls does not depend on the risk factors of the patient.

Priority:		Highest	High	Above Normal	Normal	Below Normal	Low	Lowest
High risk patient makes a	Normal call		0.2	0.6	0.2			
	Sanitary call		0.3	0.6	0.1			
	Service call			0.2	0.2	0.6		
Medium risk patient makes a	Normal call			0.3	0.6	0.1		
	Sanitary call			0.4	0.5	0.1		
	Service call				0.2	0.4	0.4	
Low risk patient makes a	Normal call				0.6	0.3	0.1	
	Sanitary call				0.7	0.2	0.1	
	Service call					0.4	0.4	0.2
Nurse treating high risk patient makes a	Assistance call	0.2	0.5	0.3				
	Sanitary Assistance call	0.3	0.5	0.2				
Nurse treating medium risk patient makes a	Assistance call		0.6	0.3	0.1			
	Sanitary Assistance call		0.7	0.2	0.1			
Nurse treating low risk patient makes a	Assistance call			0.7	0.2	0.1		
	Sanitary Assistance call			0.8	0.1	0.1		
Nurse makes a	Urgency call	1.0						
Nurse makes a	Medical call						1.0	
Nurse or device makes a	Technical call							1.0

Table B.2: The probabilistic assignment of calls to a priority category

B.2.2 Algorithms

B.2.2.1 Priority assessment of a call

The general probabilistic information in the ontology about the assignment of patients to risk groups and the priorities of calls is exploited to determine the priority of a specific call made by a specific patient as visualized in Figure B.2. Various methods have been proposed in literature to represent and reason about probabilistic knowledge in ontologies. On the one hand, approaches have been proposed based on combining some form of *Bayesian network theory* [14] with an ontology [15–17]. On the other hand, a probabilistic extension of *Description Logics* (DLs) [18] has been proposed by Lukasiewicz [19].

The latter method is easy to use and understand and offers a wide range of reasoning support. All the reasoning is done in a totally logical way without any implicit or explicit translation of the *Knowledge Base* to for example a *Bayesian network*. The methods, based on some form of *Bayesian network theory*, are more expressive and can express very difficult probabilistic dependencies. However, these methods have the limitation that both a traditional ontology and a probabilistic model have to be maintained. Logical and probabilistic reasoning has to be done separately. Because of these disadvantages and because the uncertainties needed for this use case can be expressed by the second approach, namely the probabilistic extension of DLs, is advantageously chosen as method to represent and reason with probabilistic knowledge in ontologies. An open source implementation of this method, called *Pronto* [20], was used.

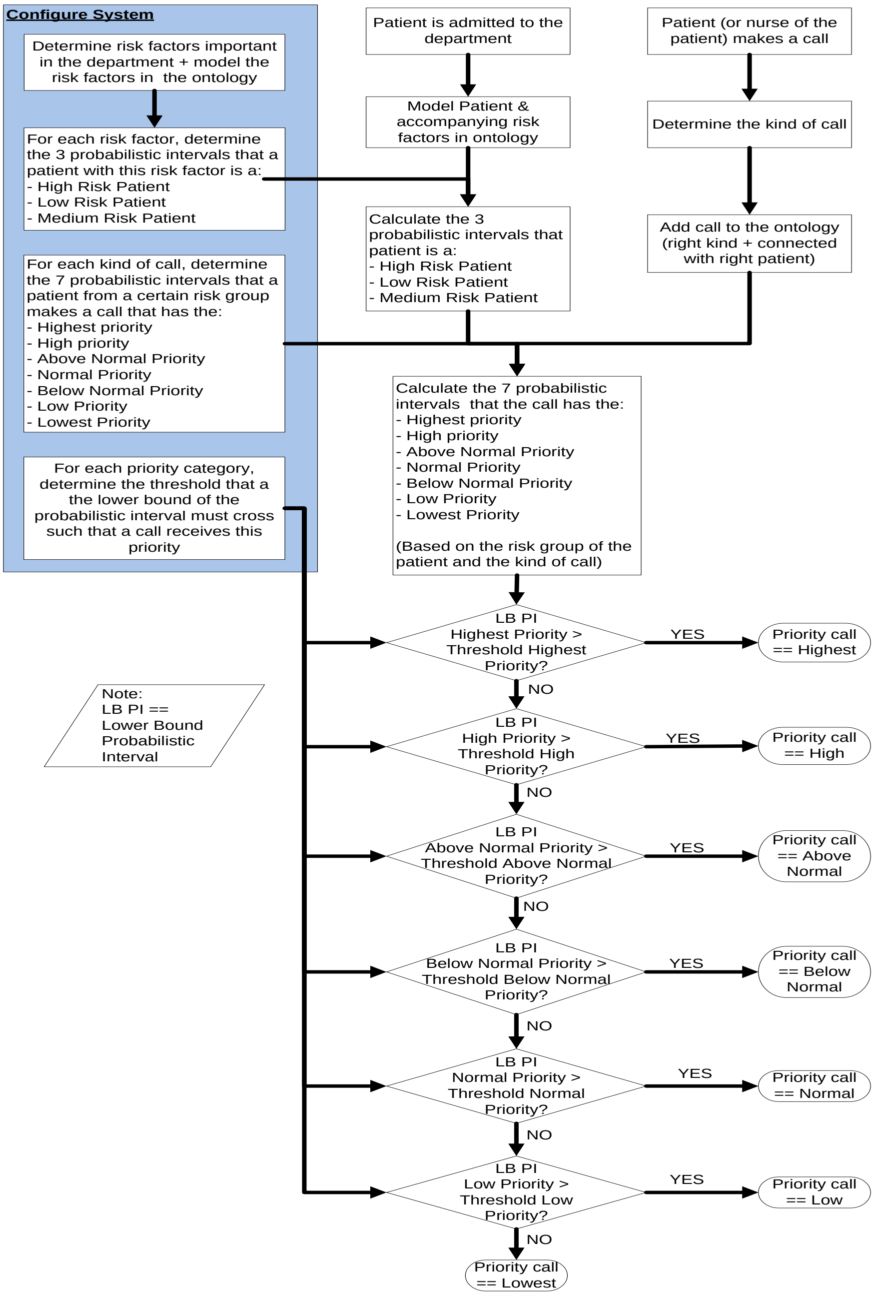


Figure B.2: Flow chart detailing the priority assessment algorithm of a call

However, some problems occurred as classical and probabilistic reasoning are not correctly combined by *Pronto*. For example, first the patient P1 is defined in the classical part of the ontology as a Heart Patient. Secondly, a constraint is added to the probabilistic part of the ontology about P1 that states that he has a specific probability of having a Neurologic Problem. However, *Pronto* does not realize that these two statements are about the same person. So when *Pronto* calculates the probability that P1 is a High Risk Patient, it does not take into account that P1 is a Heart Patient. This issue was resolved by first performing classical reasoning on the non-probabilistic part of the ontology with for example *Pellet* [21]. Secondly, the results are added to the ontology as generally true probabilistic statements, thus with interval [1,1]. Finally, *Pronto* is used to reason on all the probabilistic statements.

For each of the seven possible priorities, *Pronto* is used to calculate the probability that a specific call has this priority based on the information in the ontology. However, one priority needs to be attached to the call, so it can be used in the nurse call algorithm, see Section B.2.2.3.

To determine the suitable priority for the call based on the probabilistic values, the following threshold algorithm was employed on the lower bound of the probabilistic intervals. If the probabilistic value for the highest priority is higher than or equal to the threshold for the highest priority, it gets the highest priority. If not, the same condition is checked for High, Above Normal, Below Normal, Normal, Low and Lowest priority classes.

B.2.2.2 Determining the priority thresholds

The thresholds for each priority class for a particular hospital or department are determined by performing simulations of calls. The risk profile of the patients within this department is determined and weights are assigned to these risk factors that reflect how frequently they occur in this department. Combinations of risk factors that are deemed to be more frequent than others are also specified. For example, the risk factors *neurological problem* and *disoriented/confused* often occur together, e.g., in patients with multiple sclerosis. In some hospital departments certain combinations may be encountered more frequently than in other departments. The risk profiles used in this research are discussed in Section B.2.3.1.

Based on this collected data, 20 test groups and 10 validation groups of patients with risk factors are randomly generated, using the algorithm visualized in Figure B.3. Each group contains as many patients as there are beds within the department. Note that Flow B takes the possible combinations into account. If a risk factor is chosen then all the weights of the risk factors that it can have a combination with are doubled. Consequently, these risk factors, and thus these combinations, have more chance of being chosen. Note that it is possible that a weight of a risk factor is doubled twice if it both occurs in the general and department specific combinations. For the patients with three or more risk factors, one first needs to determine how many risk factors are going to be generated. As can be seen in the left upper corner of Figure B.3, a weighted procedure was used to make sure that a low number of risk factors is more plausible.

To determine the thresholds, each of the generated patients makes each kind of call once, i.e. a normal, service and sanitary call, and the patient's responsible nurse makes one assistance and one sanitary assistance call. The other types of calls, i.e. urgency, medical and technical, do not need thresholds as they can generally only be assigned to one priority category, namely the highest, low and lowest priority respectively.

For each call, the probabilistic intervals are calculated for each of the seven possible priorities using the probabilistic reasoning algorithms. These intervals indicate for each priority category the probability that this call has this priority. For each priority category only a limited number of probabilistic values can be obtained for the lower bound of the interval. These lower bounds are considered as possible thresholds.

For each combination of thresholds represented by the lower bounds of the calculated probabilistic intervals, it is determined for each call which priority category it gets. If the call has a probabilistic interval for the highest priority for which the lower bound is higher than or equal to the threshold for the highest priority class, the call gets the highest priority. If not, the same condition is checked for High, Above Normal, Below Normal, Normal, Low and Lowest priority classes and thresholds. If none of these conditions hold, the call gets the status Undetermined. The Below Normal threshold is checked before the Normal threshold to ensure that the default class that calls are assigned to is the Normal priority class. The Low and Lowest priority classes are generally reserved for technical and medical calls and should thus not be assigned often to other kinds of calls.

Using the above algorithm, the percentage of calls that are assigned to each priority category and the percentage of Undetermined calls is calculated for each combination of thresholds. A curve fitting algorithm is used to determine the appropriate combination of thresholds for this department. The combination of the thresholds is searched for which the percentage deviates least from the ideal distribution. Preference is given to combinations with the least amount of Undetermined calls. The ideal distribution is determined based on the characteristics of the department, e.g., frequency of calls. For example, the priority distribution 5% - 10% - 25% - 35% - 25% - 0% - 0%, ordered from the Highest to the Lowest priority, reflects a realistic hospital environment. The tested kinds of calls generally do not have the Low or Lowest priority as these categories are preserved for medical and technical calls. The middle categories, namely Above Normal, Normal and Below Normal, generally contain more calls as most calls are made for simple requests.

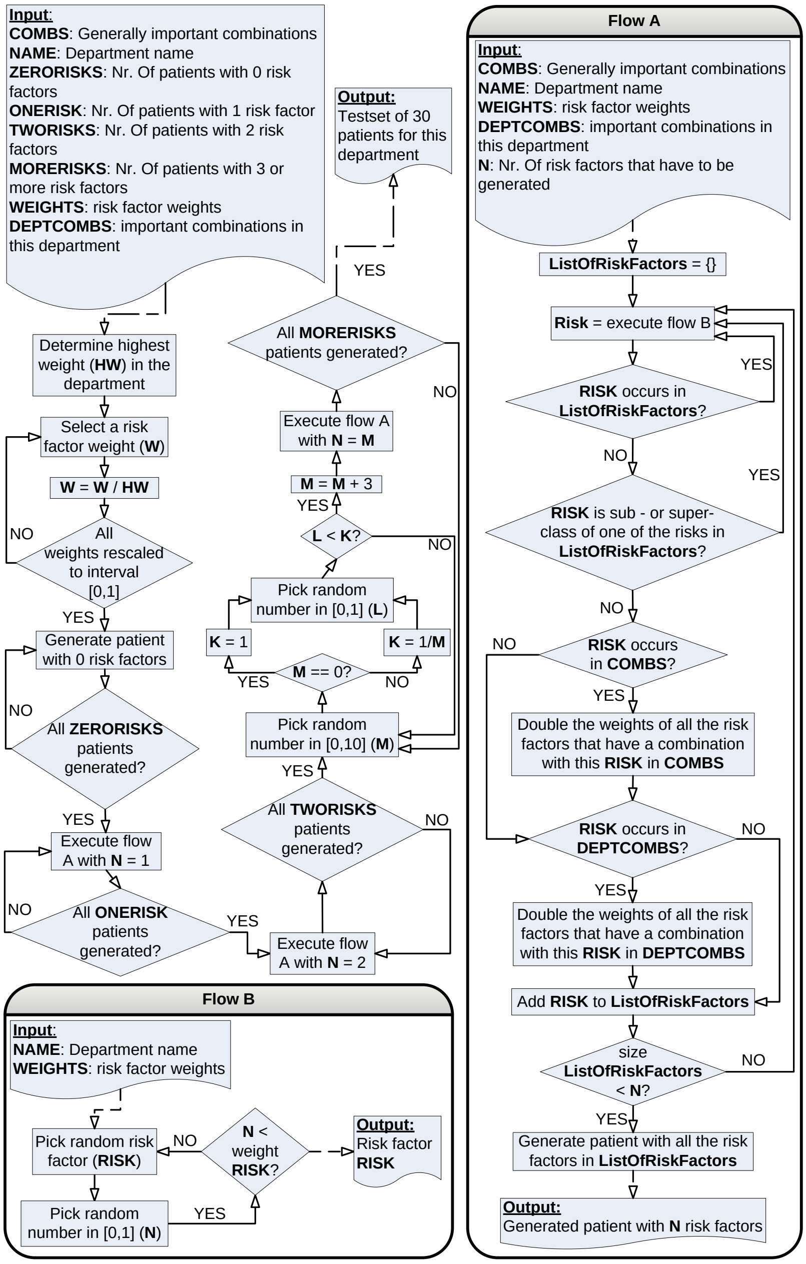


Figure B.3: Flow of the generation of a test or validation group of each department used to determine the thresholds for the priority assessment algorithm

B.2.2.3 Nurse call algorithm

The previous nurse call algorithm, which was detailed in [10], was updated to take the priorities of the calls into account. Additionally, algorithms were devised for the service, sanitary, (sanitary) assistance, technical and medical calls.

The algorithm starts with determining which kind of call has been made and acts accordingly. Normal, sanitary, service and (sanitary) assistance calls employ the same basic algorithm. The difference is that for normal, sanitary and (sanitary) assistance calls only nurses can be called. For service calls caretakers can also be called. It is also made sure that the nurse that made the (sanitary) assistance call, is not called again.

This basic algorithm first checks if the responsible nurse or caretaker, further described as helper, is in the vicinity. This responsible helper is called if he/she is busy with a task that has a lower priority than the current call.

If this responsible helper cannot be called, all the helpers who work on the department where the call originated are considered. From this group, the helpers who are not willing or qualified to treat the patient or are not in the vicinity are removed. Only for calls with the highest or high priority, helpers are considered that are busy with a task with a lower priority. Otherwise, the helpers are never able to finish the work for the patients they are responsible for. For calls with lower priorities, the busy helpers are filtered. From the remaining helpers, the one who has the most characteristics in common with the preferences of the patient is chosen. If there are no characteristics specified, the helper closest to the patient is chosen.

If this option still does not offer a solution, the search is widened beyond the scope of the department and helpers in the whole hospital are considered. The selection is similar to the previous paragraph, but busy nurses are never considered.

If the result is empty, this means that there are no available helpers in the direct vicinity. The distance becomes a deciding factor. So the closest helper with right properties is selected, e.g., free, willing and qualified.

If this still does not offer a solution, all the helpers in the hospital are considered and the one closest to the patient is called.

Note that the characteristics are only used to choose among different available helpers. They are never used to decide that a helper cannot handle a patient.

The algorithm has a time-out procedure. If a staff member has not indicated that he/she is going to handle the call within the time-out time that is specified for this type of call in the ontology, another staff member is selected to handle the call by running the algorithm again.

Urgency, medical and technical calls each have their own algorithm. For urgency calls, the priority lies on finding a person who is near instead of a person who is free. Therefore, the staff member closest to the origin of the call who is member of an urgency team and who is not already handling another urgency call is searched. Then the entire urgency team to which this person belongs is called. A time-out procedure is not needed here, as an urgency call is always immediately answered.

The algorithms for the technical and medical calls are straightforward because they generally have a low priority. For a medical call, the medical staff member responsible for this patient is called. For a technical call, the closest, free technical staff member is called. Both algorithms also have a time-out procedure.

Note that a staff member is sometimes called while he/she is busy with a task. It is up to this person to decide if he/she is going to interrupt this current task or not. In contradiction to the current nurse call system, the staff member knows that the new call has a higher priority than this current task. Based on these priorities the staff member can make a more funded decision. If the staff member decides to answer the call, the system automatically interrupts the current task. If the task is a call, another staff member is searched, using the algorithms above. If the task is not a call, the task is added to the list of tasks that this person must do.

The nurse call algorithm also takes into account that patients sometimes hit the nurse call button multiple times before their call is handled. When the nurse call algorithm notices that the hit originates from a button, which already launched a call that has not been handled yet, no new nurse is assigned to the call. Instead, the nurse who received and accepted the call is alerted that the patient pushed the button again. It was chosen not to increase the priority of the call in this as this would favor the impatient patients. Moreover, it would allow patient to cheat the system.

Finally, this algorithm also controls the nurse call lights and tracks the status of the call. When a call is launched, the status of the call is *Active*. If the call is made inside a room, the nurse call light outside the door is switched on, giving the caregivers of this department a visual cue that a call has been launched inside this room. Moreover, the lights give a visual indication to the staff how busy the department currently is. Buttons also light up in the room to assure the patient that the call was properly placed and registered by the system. The nurse call algorithm assigns the call and the assigned nurse receives the call on a smartphone. If the nurse accepts the call, the status changes to *Accepted*. When the nurse arrives in the room, an identification key is used to log in. The nurse call algorithm thus registers that the assigned nurse is present in the room. The status of the call becomes *Busy* and the light changes color, indicating to the other caregivers and visitors that a nurse is present. When the nurse leaves the room and logs out, the status of the call changes to *Finished* and the light switches off. If the call did not originate from inside a room, and a nurse is thus not able to log in when tending to the patient, the nurse can indicate on the smartphone that the call is handled.

B.2.3 Evaluation set-up

B.2.3.1 Collected data

To determine the thresholds for the priority assessment algorithm and to evaluate the sensitivity of these thresholds, data was gathered about the risk profile of patients in five nursing departments of Ghent University Hospital [11]. Each department can contain at most 30 patients. The data is summarized in Table B.3. The following combinations of risk factors were deemed to be more frequent than others in all departments:

- COPD and tracheotomy
- High age and disoriented/confused
- Diabetes and disoriented/confused
- Neurological problem and disoriented/confused
- Transferred from the ICU and tracheotomy

The risk factor combinations which are more frequent in specific departments are indicated in Table B.3.

To evaluate the developed probabilistic nurse call systems, simulations were performed based on realistic data gathered from three of these departments. These departments differ in the mobility of the patients. In *Dept1*, the patients are barely mobile. Most of the patients are in a coma and those that are awake are attached to a lot of equipment. *Dept2* contains patients that are fairly mobile, but spend most of their time in their room. In *Dept3*, the patients are quite young, so they move around a lot across the whole department. The floor plans of Dep1, *Dept2* and *Dept3* are visualized in Figure B.4.

The most important properties of these departments are summarized in Table B.4. For each department, at least three spaces were selected to where the patients often travel. The time it takes to travel to all these spaces from the respective departments was measured. It was also determined how patients divide their time over these different spaces. Some information about the staff in these departments was also gathered. In *Dept1* and Dept, each nurse is responsible for at most two patients who often lie in adjacent beds in the same unit. In *Dept2*, each nurse is responsible for a couple of patients. They are assigned based on the location of the rooms. Patients in rooms close to each other are often assigned to the same nurse. This often leads to an unevenly distributed workload amongst the different nurses. In none of these departments a patient is ever assigned to more than one nurse at the same time. *Dept2* and *Dept3* also employ head nurses, but he/she generally does not answer calls. As urgency calls can be made in *Dept1* we also have to take the doctors into account that can take part in the urgency teams that

Departments:		Dept1	Dept2	Dept3	Dept4	Dept5
Number of patients with	0 risk factors	0	10	15	4	0
	1 risk factor	9	10	10	10	2
	2 risk factors	6	8	4	12	8
	> 2 risk factors	5	2	1	4	20
Risk factor weights (%)	High age (a)	36	50	0	15	55
	Diabetes (b)	8	10	20	2	5
	Heart disease (c)	10	3	19	2	5
	High fall risk (d)	2	5	0	13	5
	Neurologic problem (e)	7	3	10	16	5
	Tracheotomy (f)	5	10	5	5	2
	COPD (g)	10	3	1	2	5
	Paraplegia (h)	2	3	2	16	3
	Pneumonia (i)	10	3	30	2	5
	Disoriented/ confused (j)	5	5	0	2	5
	Gastric Bleeding within 48h (k)	3	3	1	2	2
	Transferred from ICU (l)	0	1	10	15	2
	Transferred from ICU within 72h (m)	0	0	5	0	0
	Reanimated (n)	2	1	2	8	1
	Reanimated within 72h (o)	2	0	0.05	4	0
Combinations		f and g a and c	a and d b and j	None	d and h j and n j and o	a and d

Table B.3: The specifics of the five departments for the threshold research

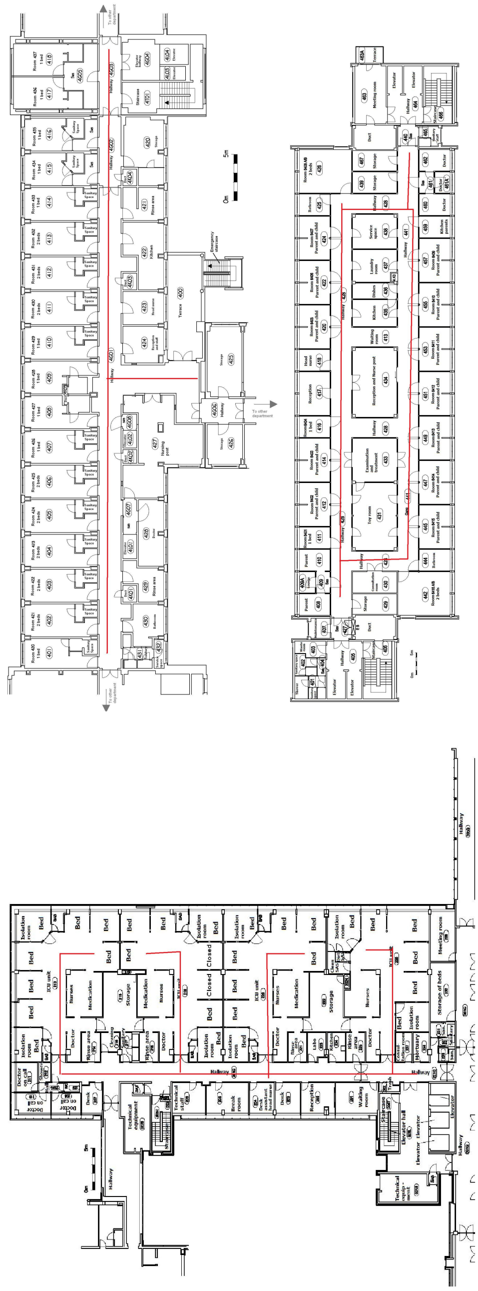


Figure B.4: The floor plan of Dept1, Dept2, Dept3

Properties:			***Dept1***	***Dept2***	***Dept3***
Mobility			Low	Medium	High
Nr. of beds			22 adults, 6 children	26	18
Occupation rate			116.15%, 87,58%	84.62%	70%
Most visited spaces by patients			Angiography room, CT scanner, Radiology department, Recovery	Smoking area outside, Cafeteria, CT scanner	Surgery, Ultrasound, Receiving electrodes to monitor sleep
Nr. of nurses	**Early shift**	**Week**	14	5	3
		Weekend	14	4	Closed
	Late shift	**Week**	14	4	2 or 3
		Weekend	14	3	Closed
	Night shift	**Week**	14	1 or 2	1
		Weekend	14	1 or 2	Closed
Nr. of head nurses (who answer calls)	**Week**		1 & 2 assistants	/	/
	Weekend		1 assistant	/	/
Nr. of doctors (who answer urgency calls)	**Day shift**		3 interns 1 pediatrics intern 1 or 2 residents 1 pediatrics resident	/	/
	Night shift		1 intern 1 pediatrics intern	/	/
	Weekend		/	/	/

Table B.4: The specifics of the 3 departments of Ghent University Hospital used for the simulations of the probabilistic nurse call system

handle these calls. The number of available doctors is summarized in Table B.4. On the floor above *Dept1* there is always a doctor available who can be called in critical situations. All the present nurses and doctors are able to handle and receive urgency calls.

The walking behavior of the staff was simulated by gathering information about their tasks and the percentage of time they spend on each group of tasks [22], as visualized in Figure B.5. Each of the tasks was also assigned a priority based on how easily the task can be interrupted.

B.2.3.2 Currently employed nurse call algorithm

Normal calls can be made in all the departments and sanitary calls in *Dept2* and *Dept3*. In *Dept1* sanitary (assistance) calls cannot be made as all the patients either have a catheter or a bedpan. Service calls cannot be made as the Ghent University Hospital does not employ caretakers. Nurses are able to make (sanitary) assistance calls in departments *Dept2* and *Dept3*, but only *Dept1* has buttons to make urgency calls. Medical calls cannot be made in any of these departments.

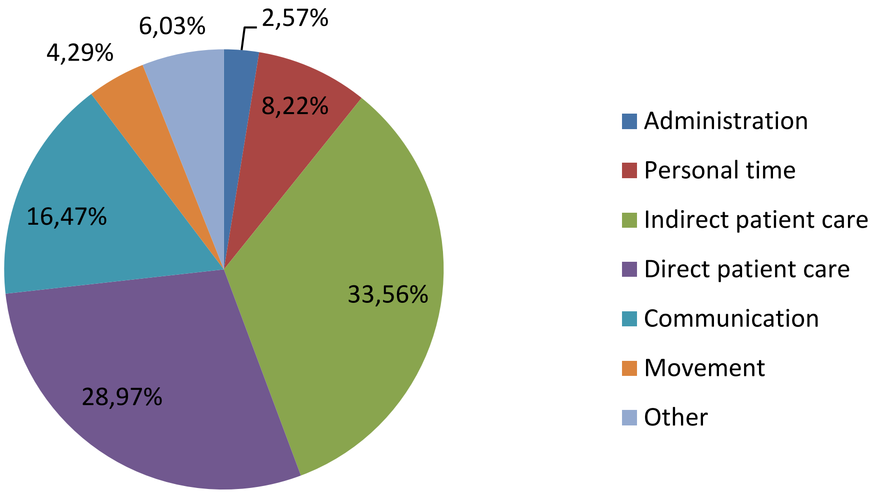


Figure B.5: Distribution of time of the nurses across different kinds of tasks

Technical calls are not taken into account in the following simulations as they always get the lowest priority and a member of the technical staff is called as explained in Section B.2.2.3.

Logging information, such as frequency and duration, about the calls in the departments was gathered during three weeks. Limited research has been done about reasons for patients' call light use in the Ghent University hospital. The paper by Meade [23] presents an extensive study about this subject. The used results are presented in Figure B.6. When a call is made, a reason is assigned based on these percentages. The average time that a nurse spends on handling a task from each category was also determined.

Calls are currently handled differently in the three departments. In *Dept1* the nurses and doctors do not carry around beepers or portable phones. When a patient makes a normal call, a light shines above his/her bed and a buzzer starts making noise. A nurse who is in the neighborhood will tend to the patient. If he/she is currently busy and there is nobody else around, the nurse has two options. In the first case, he/she turns off the light and buzzer, finishes his or her current task and then moves on to this call. In the second case, the nurse first handles this call and then goes back to his or her current task. In both cases, the nurse has to remember him- or herself to go back to either the call or the current task. Assistance calls are handled in a similar manner by a second nurse who is also in the neighborhood. When an urgency call is made, a loud alarm is sound throughout the entire department and a bright light is lit above the bed of the patient. All nurses hurry to the origin of the alarm, but it is made sure that at least one nurse stays behind on every

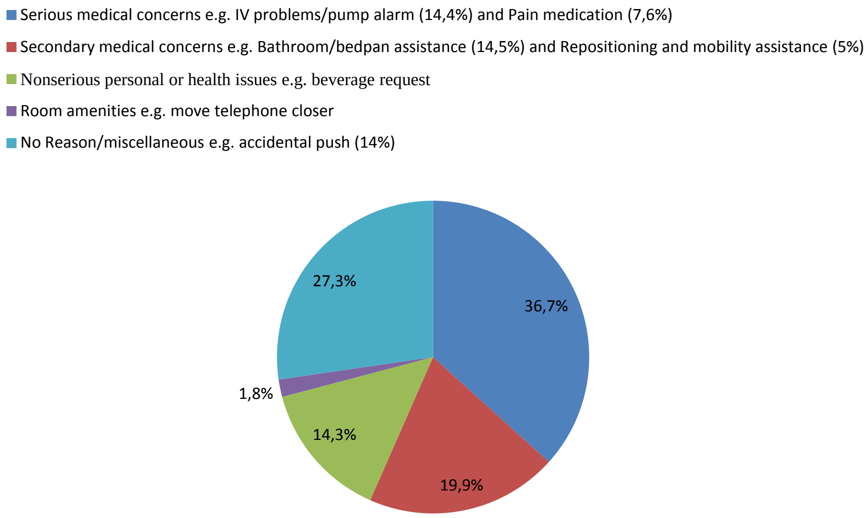


Figure B.6: Reasons for patients' call light use

unit of the department. All the doctors attached to this department, also receive the urgency call on their portable phone or beeper. If they are available, they rush to the patient. There is always a doctor present who always responds to these calls.

In *Dept2* and *Dept3* the nurses receive the calls of all the patients in the department on their beeper or portable phone. A light switches on above the room of the patient. The nurse, who arrives first in the room, switches off the call and treats the patient. Sanitary and (sanitary) assistance calls are handled similarly. If the time-out of a call is reached and none of the nurses have handled the call, all the nurses are called again. It is possible that multiple nurses arrive at a room to handle a call, as multiple nurses are called and one nurse does not know if another nurse will handle the call or not. If they interrupt their current task (which could also be a call) to handle this call, the nurses have to remember themselves that they have to go back to that interrupted task. In case of an interrupted call, the other patient also has to wait until the nurse has finished this call, while it could be of a lower priority.

B.2.3.3 Implementation and simulation set-up

The *oNCS*, incorporating the previously described algorithms, was built as an extension of the Context-Aware Service Platform, CASP [24, 25]. A detailed description of how this extension was realized can be found in Ongenae, et al. [12].

A mobile nurse call application was also developed. It is used by the caregivers to receive, assess and accept, i.e., indicate that they are going to handle, calls. When the patient hits the call button multiple times, this is indicated by the application. A nurse can also use the application to contact the patient, e.g., to request the reason for the call or to give feedback to the patient about the expected arrival time, or other staff members. This application is further explained in Ongenae, et al. [5].

Two scenarios were simulated for each department, namely a realistic Poisson scenario and a worst-case scenario. Only the Poisson scenario is discussed in this paper as conclusions made for the Poisson scenario also apply to the worst-case scenario.

It is assumed that the patients possess portable buttons and can thus move around freely and make calls. When this situation is simulated for the current system, some calls may be impossible to handle, e.g., calls made in the middle of a hallway. The movements of the patients and nurses were determined out of the collected data, as discussed in Section B.2.3.1. Once a patient makes a call, it is assumed that he/she stands still. If a nurse has to choose between multiple calls to handle, he/she chooses the one with the highest priority. If the calls do not have priorities, which can occur in the current system, or multiple calls have the same priority, the closest call is chosen. No urgency calls can be made in these scenarios. Patients or nurses who are on the move advance 1 meter in the direction of their goal during each time step. Characteristics of patients and nurses, risk factors of the patients and responsibility of staff members for particular patients were simulated as indicated in Section B.2.3.1.

In the Poisson scenario, a realistic day-to-day hospital scenario is simulated. The beds in the departments are occupied averaging around the occupation rate indicated in Section B.2.3.1. Patients can make calls modeled according to a Poisson process with λ=0.000388558 in *Dept1*, λ=0.001164021 in *Dept2* and λ= 0.000112434 in *Dept3*. Nurses receive calls of patients while they are doing their tasks. They will only interrupt their current task, if the call has a higher or equal priority. They will only interrupt current calls, if the new call has a higher priority. If the new call does not have a priority, a nurse chooses randomly to interrupt the current task or call. During the handling of a call, they will launch a (sanitary) assistance call with a probability of 0.02979% in *Dept1*, 0.07386% in *Dept2* and 0.20588% in *Dept3*. Each simulation was done 30 times for each of the shifts during the weekend and week. The simulations performed done on a PC with an Intel Core 2 Duo P8600 processor and 4 gigabyte RAM.

B.3 Results

B.3.1 Threshold parameter exploration

This section explores the sensitivity and applicability of the thresholds generated by the algorithm detailed in Section B.2.2.2. This algorithm was used to randomly generate patients, calls and their accompanying probabilistic intervals based on the data gathered about the risk profiles of patients in five departments of the Ghent University Hospital as described in Section B.2.3.1. In total 22,500 calls were simulated and 252,252 possible combinations of thresholds, represented by the lower bounds of the calculated probabilistic intervals, were achieved. To evaluate the fluctuation of the total deviation, the 100 threshold combinations with the smallest total deviation from the ideal distribution, namely 5 - 10 - 25 - 35 - 25 - 0 - 0, were studied. The 100th one has a deviation of 23.93 and 24.04 in the test and validation group respectively. This shows that the deviation does not increase fast.

There were 12 combinations of thresholds which had the smallest total deviation, namely 16.44 and 19.20 in the test and evaluation group respectively, from the ideal distribution. Each of these combinations had no Undetermined calls. Combinations are preferred that do not assign the Highest priority too much, as it is mainly used for the urgency calls. Six of the 12 threshold combinations only assigned around 5% of calls to this priority. The difference between these combinations is the threshold for the Above Normal (0.21, 0.24 or 0.3) and Low (0 or 0.2) priority. As the Normal priority class has a threshold of 0 for each of the six combinations, the calls never get the Low or Lowest priority. So 0 is picked as threshold for the Low priority. For the Above Normal priority, the middle threshold of 0.24 is chosen as this guarantees that it is less sensitive to fluctuations.

Thus, the chosen thresholds are 0.21 - 0.3 - 0.24 - 0 - 0.05 - 0 - 0, ordered from the Highest to Lowest priority.

The sensitivity of the chosen thresholds was studied by selecting the threshold for one priority, e.g., the Highest, and increasing and decreasing it to the next threshold for this priority, e.g., 0.24 and 0.18. The fluctuation in deviation for these threshold combinations is illustrated in Figure B.7a. The percentage of calls that changes priority by changing the threshold is shown in Figure B.7b. The x-axis shows for each priority to which threshold the chosen threshold is increased and decreased, while the rest of the thresholds stay the same as in the chosen combination. As the chosen threshold for the Normal and Low priority is 0, it can only be increased. The Lowest priority had only one threshold possibility, thus it cannot be changed. As can be derived from the graphs, only the Normal and Below Normal priorities are very sensitive to threshold changes. This is because the Normal priority is used as the default priority with a threshold of zero. All the calls for which the probabilistic value is equal or lower than the thresholds for the Highest, High, Above and Below Normal priorities, are assigned the Normal priority. Consequently, the Low priority becomes the default priority, when the threshold of the Normal priority is increased. This causes a lot of calls to receive this priority instead of the Normal one. If the threshold of the Below Normal priority is decreased, it becomes zero and this priority becomes the default.

The chosen thresholds were also evaluated by studying which percentage of each kind of call and which percentage of patients with zero, one, two and three or more risk factors are assigned to each priority. The results are illustrated in Figure B.8. The normal and sanitary calls mostly get the Above Normal and Normal priorities. A low percentage of these calls get the Below Normal priority. Service calls get generally a lower priority. This is realistic as these calls are made for caring tasks. The (sanitary) assistance calls often get the Highest or High priority. This is desirable as immediate help is often required for a nurse in these situations. The calls of patients who have no risk factors are assigned the default priority, namely Normal, as risk factors are the only criteria used in this initial study to determine the priority. Calls made by patients with one risk factor are distributed equally among all the priorities. In this case, the priority depends on the severity of the risk factor. As the number of risk factors increases, the calls are more likely to receive a higher priority.

B.3.2 Simulation results

Figure B.9 shows for the three departments the number of calls that have a nurse present before a specific time point. Note that the first part of the x-axis goes in jumps of five seconds, while the second part jumps 60 seconds each time.

In *Dept1* there is no significant difference between the *oNCS* and current system as a lot of nurses are present who can readily answer calls.

However, in *Dept2* and *Dept3* there is a notable difference between the two systems. In the *oNCS* only one nurse receives the call, which often has a higher priority than his/her current task. Therefore, the nurse immediately answers the call. In the current system, multiple nurses receive the call. They decide if they quit their current task without additional context information. If all the nurses ignore the call, it has to be relaunched, as illustrated by peaks on the graph after the relaunch times, e.g., 180 or 360 seconds. Moreover, the *oNCS* takes the walking distance into account when assigning a nurse. The distance to the patient is thus limited. This explains the difference between *Dept2* and *Dept3*. The circular design of *Dept3* makes it possible for the nurses to reach rooms faster.

In all three departments, the peak of the *oNCS* occurs somewhat later. Running the nurse call algorithm causes an initial delay.

In *Dept1* and *Dept2*, the tail of the *oNCS* is very long, as all impossible calls are answered. Most of these calls occur in distant places. This could be compensated by allowing nurses from closer departments to answer these calls.

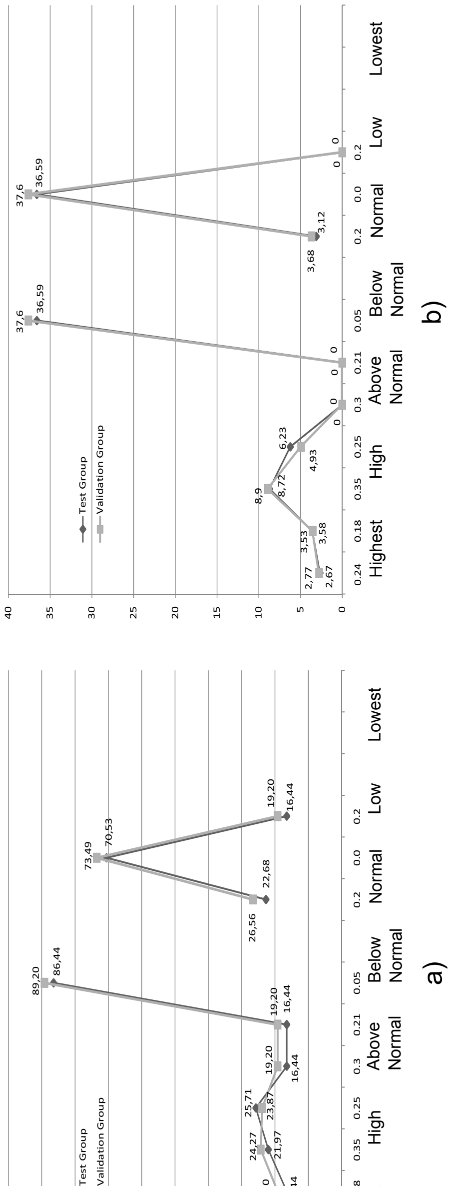


Figure B.7: Study of the sensitivity of the threshold parameters: a) Fluctuation of the deviation b) The percentage of calls that changes priority category

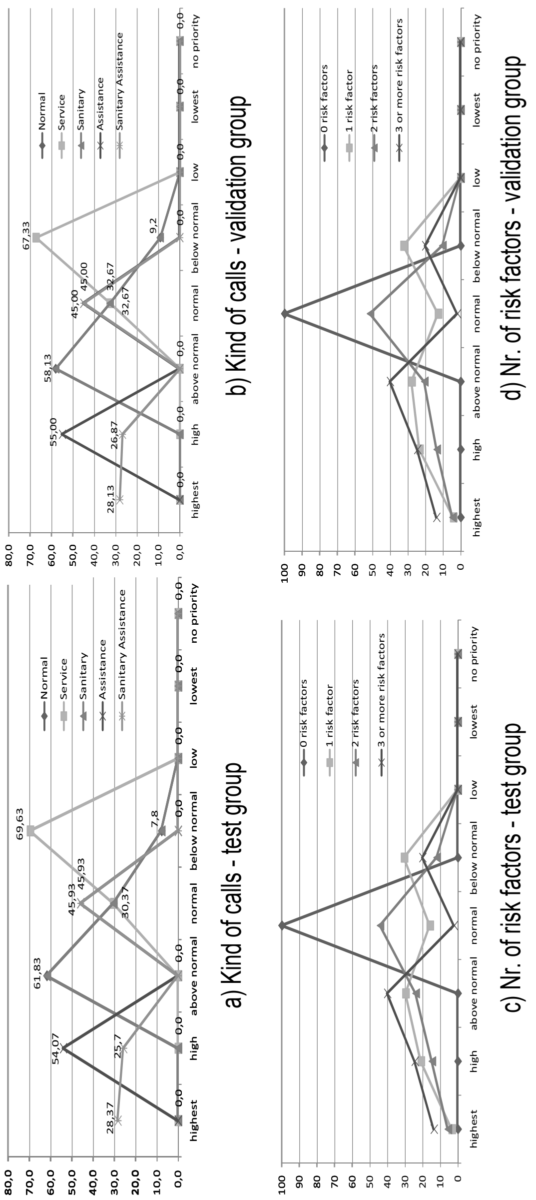


Figure B.8: The percentage of each kind of call and the percentage of patients with 0, 1, 2 and 3 or more risk factors that are assigned to each priority category, both for the test group as the validation group

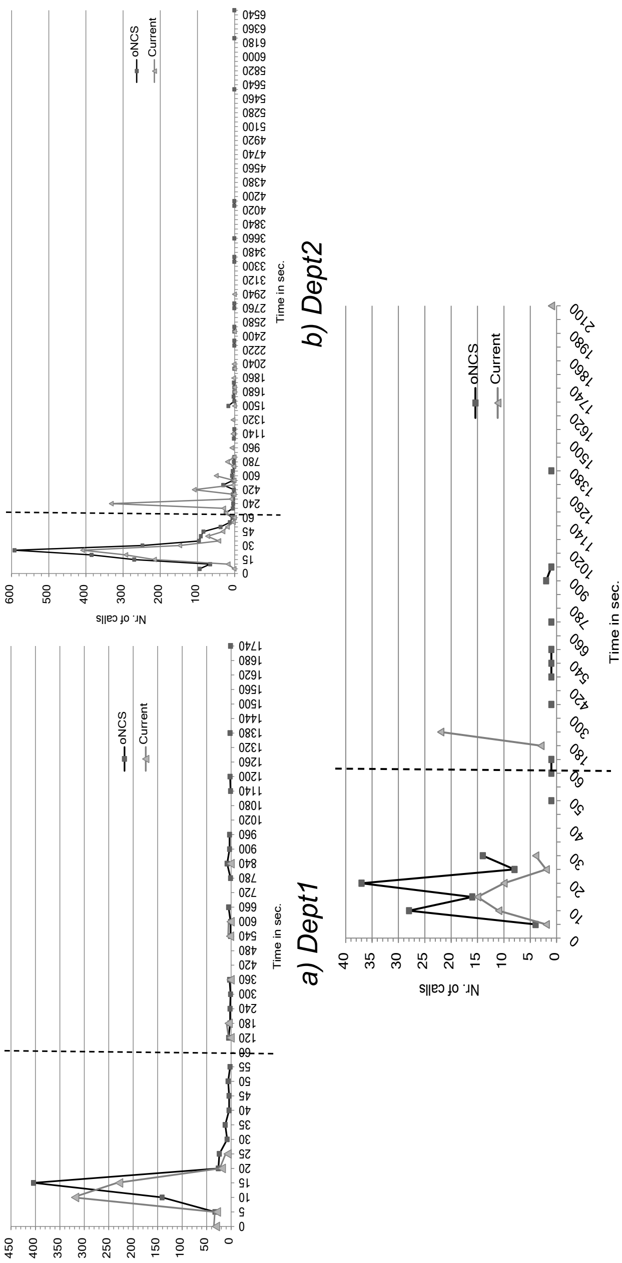


Figure B.9: The number of calls (y-axis) that have a nurse present before a specific time point (x-axis, in seconds)

Figure B.10 visualizes the percentage of calls of a particular kind that have a nurse present before a specific time point.

In *Dept1*, the curves of the assistance calls are comparable between both systems, as these calls generally have a very high priority. The normal calls generally have a nurse present faster in the current system. This can again be explained by the high amount of nurses present in the department. In the current system every nurse that sees the call can handle it. In the *oNCS* only one nurse receives the call on his/her beeper and is expected to handle it. Moreover, the nurse that is called in the *oNCS* might not necessarily be the closest free nurse, as the algorithm prefers to call the responsible nurse.

In *Dept2*, the sanitary assistance calls have a nurse present within 15 seconds in both systems. However, the *oNCS* performs much better than the current system for the other kinds of calls. Due to the lower amount of nurses in this department, an occupied nurse is often called. The *oNCS* prefers occupied nurses who are busy with a task that has a lower priority than the call. This is often the case for assistance and sanitary calls. As the nurse is aware of this higher priority of the call, he/she immediately handles it. In the current system, the nurse has to decide at random to answer the call or finish the task first.

In *Dept3*, the curves of the assistance calls are quite different between both systems. The curve of the *oNCS* rises faster but then slows down and has a much longer tail. Sanitary calls on the other hand are handled very fast in the *oNCS*. However, only a few sanitary calls were simulated. In *Dept3*, the toilet is inside the room. Calls made inside the room were thus registered as normal, as it is impossible to track with enough accuracy if the patient is on the toilet or not. Only calls made inside designated sanitary areas were registered as sanitary. This explains why no sanitary assistance calls were simulated. The normal calls generally have a nurse present faster in the *oNCS*. One kind of call does not get answered notably faster than another kind in *Dept3*. As patients in this department have a wide variety of risk factors, a normal call can have the same priority as for example an assistance call. Consequently, the assistance call is not always handled faster than the normal one.

The number of calls with a particular priority that have a nurse present before a specific time point are visualized in Figure B.11.

In all departments, the highest priority is assigned the least, which is desirable as it is mostly reserved for urgency calls. In *Dept1*, the above normal priority is assigned the most, as *Dept1* contains a considerable amount of patients with two or more risk factors. In *Dept2* and *Dept3*, the below normal priority is assigned the most as these departments contain a lot of patients without risk factors.

In the *oNCS*, the highest priority calls are handled the fastest in all the departments. In *Dept3* two assistance calls with the highest priority were made while all the other nurses were away, causing these calls to be handled quite slowly. The be-

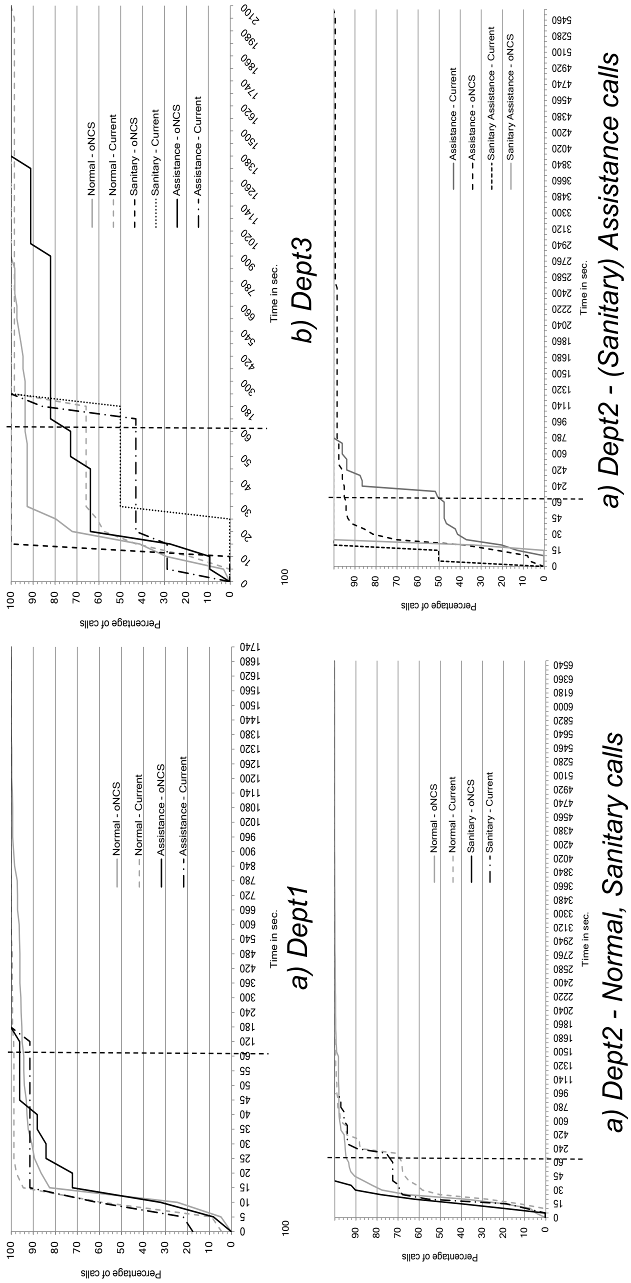


Figure B.10: The percentage of calls (y-axis) of a particular kind that have a nurse present before a specific time point (x-axis, in seconds)

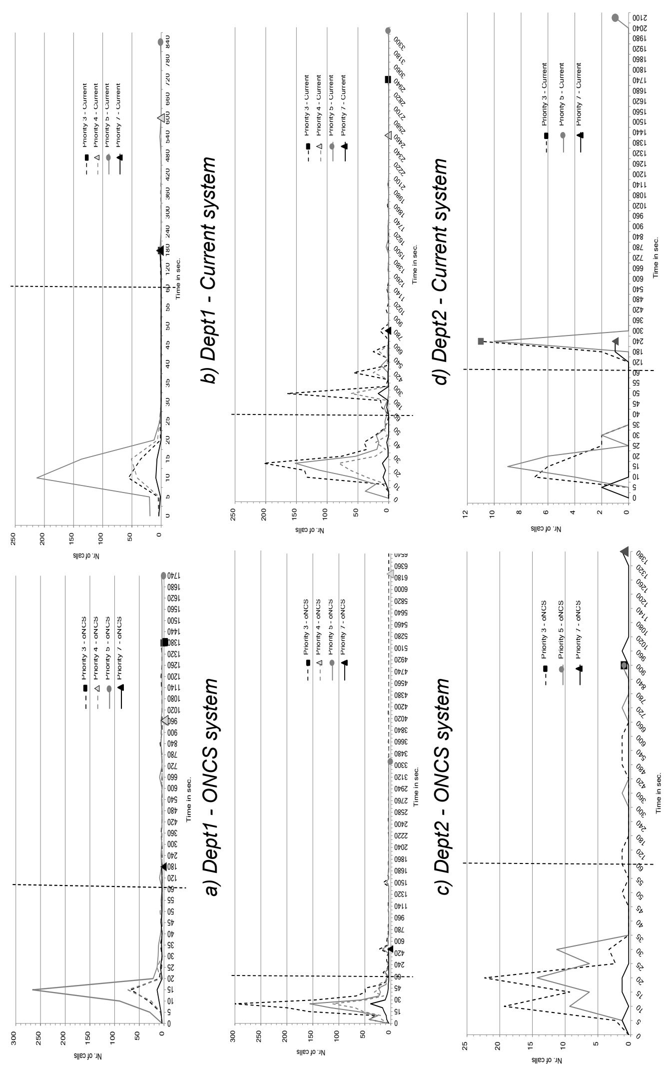


Figure B.11: The number of calls (y-axis) with a particular priority that have a nurse present before a specific time point (x-axis, in seconds)

Department	Nr. of nurses in the department	Workload distribution current system Max.	Min.	# 0%	Std. Err.	Workload distribution *oNCS* Max.	Min.	# 0%	Std. Err.
Dept1	15	44.4	0	241	9.0	40	0	206	7.6
	17	42.9	0	265	7.5	40	0	252	7.3
Dept2	1	100	100	0	0	100	100	0	0
	2	71.0	29.0	0	11.9	57.1	42.9	0	5.1
	3	58.3	12.1	0	12.7	61.0	7.0	0	14.7
	4	62.5	0	1	12.3	50	0	1	10.2
	5	54.8	0	2	12.6	42.5	2.6	0	7.9
Dept3	1	100	100	0	0	100	100	0	0
	2	100	0	2	47.4	75	25	0	17.0
	3	100	0	20	41.3	100	0	10	32.6
	4	100	0	17	32.0	100	0	8	23.2

Table B.5: Shows the distribution of calls amongst the nurses for each department and for each possible number of nurses in this department. The maximum and minimum percentage of calls assigned to a nurse and the number of nurses who got no calls is shown. Finally, the standard error is given between the percentage of calls that nurses handle and the mean. The mean is the ideal percentage of calls that a nurse should handle.

low normal, normal and above normal priority calls are handled somewhat slower than the highest priority calls. In *Dept3*, calls with these priorities also have the same worst-case time. In *Dept2*, the tails of the worst-case time are in the correct order: first the above normal, than the normal and finally the below normal priority calls. However, in *Dept1* the above normal priority calls have the highest worst-case time, as many calls are assigned this priority. If a lot of these calls are made simultaneously, they interfere with each other.

It is clear that the current system does not take the priorities into account. Even the highest priority calls need to be relaunched a couple of times. The above normal, normal and below normal priority calls have the same trend in *Dept2* and *Dept3*. However, the improvement of the *oNCS* compared to the current system is not as notable in *Dept1*. Although, the below normal priority calls are generally handled slightly faster than the normal priority calls in the current system, the rest of the graph is quite similar to the graph of the *oNCS*. This is caused by the high amount of nurses in the department.

Finally, the distribution of calls amongst the nurses is illustrated in Table B.5. The first column indicates the number of nurses in the department during the simulation. The maximum and minimum percentage of calls handled by a nurse during a shift, how many nurses handle zero calls during a shift and the standard deviation between the percentage of calls that nurses handle and the mean are indicated.

The *oNCS* leads to a slightly better workload distribution in *Dept1* and *Dept2* and a much better result in *Dept3*. There are less nurses that get the extreme percentages and the highest percentage of calls that a nurse can be assigned is also lower. In *Dept3*, the improvement is especially notable in case there are only two nurses in the department. In the *oNCS*, all the nurses get calls, while in the current system only one of the nurses handles the calls.

B.3.3 Performance results

B.3.3.1 Performance of the probabilistic ontology reasoning

To evaluate the scalability of the probabilistic reasoning, reasoning is performed on an ontology with gradually increasing number of probabilistic statements. First, the consistency and satisfiability, i.e., having a probabilistic model, is checked. Next, some probabilistic statements are entailed on concept level that were and were not explicitly stated in the ontology. For the first, the stated probabilistic interval is returned, while for the latter *Pronto* needs to reason on the ontology. Finally, some probabilistic statements on instance level, which were explicitly stated, were entailed.

The results are visualized in Figure B.12. For each task, time gradually increases and starts exploding around 20 probabilistic statements. At 24 statements, the performance becomes unacceptable, namely around 30 or 40 minutes. The scalability of *Pronto* is thus an issue [26]. However, performing the probabilistic reasoning tasks on 12 or less probabilistic statements is always below 4 seconds, which is acceptable for our application.

The following optimization was employed to speed up the probabilistic reasoning. First, during down-time, the probabilistic values for each patient that he/she is a high, medium or low risk patient are calculated and stored as known facts in the ontology. This procedure does not have to be repeated often for a patient as risk factors do not change often for a particular patient during his/her stay at the hospital. Next, when a call is made, all the probabilistic statements needed to calculate the priority of this call are extracted from the ontology. This will be at most 12 statements, namely the statements about the probabilistic assignment of this patient to the risk groups and the generic probabilistic assignment of this kind of call to the priority groups.

B.3.3.2 Performance of the nurse call algorithm

When a call is launched, a suitable nurse is notified within 50.3 ms on average, which is a negligible delay. Note that these results do not take into account the probabilistic reasoning to determine the priority of the call. As mentioned in the previous section, this reasoning is done beforehand. A more thorough discussion of this result can be found in Ongenae, et al. [12].

B.4 Discussion

First, maintaining the profile information leads to a lot of advantages. The novel nurse call algorithm intelligently assigns nurses to calls based on this information, e.g., the characteristics and the status of the staff members, the risk factors and

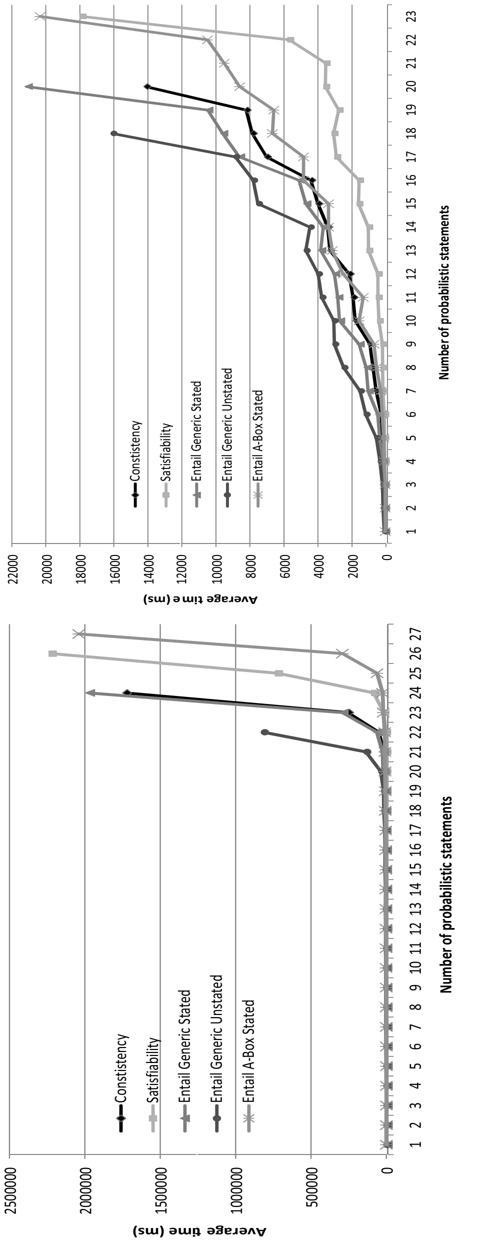


Figure B.12: The performance results of the probabilistic reasoning tasks preferences of the patients and the priority of the call. Moreover, nurses that are too far away are not called to handle a call. The currently employed nurse call algorithm only looks at which patients are allocated to which nurses. A nurse is thus sometimes called when he/she is very far away.

Additionally, in the *oNCS* the nurse knows exactly which patient made the call and the patients' current location. This can be used to determine whether to answer to call. The nurse can also directly contact or check the condition of the patient on the mobile application and thus assess if he/she needs to bring equipment or medication. Even when the patient does not make a call, the nurse can access useful information about the patients, e.g., their risk factors or room numbers.

In the current system unnecessary nurses are often called, resulting in multiple nurses (or no nurse at all) arriving at a room to handle the call. This leads to interruptions of other tasks and unnecessary walked miles. Only one nurse is called in the *oNCS* to handle a call. If this nurse accepts the call, it is also expected that he/she will tend to it. Other nurses are thus not aware of the call unless they see the light above the room of the patient. The *oNCS* thus also requires a change in mentality from the caregivers pertaining to how calls are handled. The nurses need to trust that the system has assigned the nurse who is most appropriate at that moment. They thus need to let go of the instinct to walk into a room because the light is burning. This allows them to focus on their, possibly more urgent, current task. However, they are able to walk in and start tending to the patient if they want to.

The nurse can also use the mobile application to contact other staff members or view information about them, e.g., their location or current task. This makes it easier for the nurse to determine whether to answer a call or not. Additionally, it has been shown that a lot of time in hospitals is spent on trying to find someone. This could be reduced by using the *oNCS*. Nurses can also indicate that they are going to answer a call.

However, the current mobile application does not allow to see an overview of all the current calls and which nurses are assigned to them. So it is difficult to get in touch with the assigned nurse to notify that somebody is already handling the call. A page could be added to the mobile application where all the current calls are listed with their status and who has been assigned to them. This could easily be achieved as all this information is readily available in the ontology.

Finally, when a task is interrupted, the nurse does not have to remember to return to it. The *oNCS* does this for the nurse. This leads to fewer forgotten tasks and less work pressure on the staff.

Second, the novel nurse call algorithm also leads to significant measurable improvements in the manner nurses are assigned to calls. Generally a better workload distribution amongst the nurses is achieved as the algorithm takes the current task of the nurse and its priority into account. Additionally, only one nurse is called to handle a call and the distance is taken into account when a nurse is selected. As a consequence, patients are generally treated quicker than in the current system. The time to intervention is an important parameter as it is important to quickly assess the situation when a call is made. The health of the patient could be compromised in which case fast intervention is of paramount importance. Some countries even outline general guidelines that stipulate that the time to intervention should be within 3 minutes when an urgency call was made and within 5 minutes for other calls. The time intervention is also an important parameter correlated with patient satisfaction [27] and job satisfaction of the nurses. The latter is because a quicker time to intervention leads to fewer calls being assigned again because of a time-out and thus to less unnecessary work interruptions for the nurses. Work interruptions are one of the main factors of cognitive fatigue and errors and have a significant impact on the workload distribution and performance [28].

It was also shown that the novel nurse call algorithm takes the kind and priority of the call into account. Calls with a higher priority are generally handled faster than calls with a lower priority. This is not the case in the current system. Moreover, (sanitary) assistance calls are also generally handled faster than normal and sanitary calls. This is achieved because when a nurse receives a call while performing a task, the nurse is sure that the new call has a higher priority. This way the nurse can make a more well-funded decision on whether to interrupt the current task. Moreover, the nurse is more likely to interrupt the task as the nurse knows that this call has a higher priority and he/she is the most appropriate nurse to handle this call at this moment.

As the results of the simulations of *Dept2* and *Dept3* clearly show, the benefit of the *oNCS* with probabilistic priority assessment is biggest in nursing units with a small number of caregivers, who answer calls, compared to the number of patients and a high degree of patient heterogeneity, meaning that the patients have very different risk factors and the department thus contains low, medium and high risk patients. The first is clearly illustrated by the simulations in *Dept1*. In this department there is one nurse per two patients. Up to six beds are also grouped in the same space. Patients thus do not lie in separate rooms. Consequently, there is always a nurse close to the patient making the call. Moreover, this nurse can also easily view and talk to this while he/she is busy with another patient. This means that in *Dept1*, the walking distance to the patient making the call and the need to interrupt tasks for calls is far less than in *Dept2* and *Dept3*. This conclusion is also reflected by the workload distribution, which stays more or less the same when comparing the *oNCS* to the current system.

The second is caused by the fact that patients with a similar risk profile will make calls of similar priority. This is clearly illustrated by the simulations of *Dept1*. Most of the patients in this department have a lot of risk factors, which causes that most of these patients are classified as high risk patients. Thus if patients with the same risk factors (or similar ones) make calls, the priority of these calls only probabilistically depends on the kind of call. As no differentiation can be made based on the priority of the call, the nurse call system will try to find caregivers who are free and/or close to handle the call.

Thus, in these cases the probabilistic priority assessment algorithm contributes little to the nurse call assignment. However, as previously discussed, the *oNCS* still offers a lot of other benefits that are still applicable in such departments.

The issue could be addressed in future iterations of the system by letting the priority of the call depend on more or other information than just the risk factors of the patient and the kind of call. For example, the specific medical parameters monitored about the patient at the time of the call, e.g., high temperature or blood pressure, or the profile information of the patient, e.g., frequent caller or not, could be taken into account. Moreover, before the *oNCS* is installed in a nursing unit, it needs to be evaluated if the probabilistic priority assessment, which requires considerable computation cost, will have significant impact on the nurse call assignment, i.e., enough heterogeneity amongst the patient and a small number of nurses compared to the number of patients. If not, the *oNCS* could be installed with a simpler nurse call algorithm, thus offering all the benefits of the context-awareness and portable buttons without the computational cost of the probabilistic priority assessment.

The performance of the novel nurse call algorithm is also very good, as a suitable nurse is notified within 50.3 ms on average, which is a negligible delay.

The system scales up to at least 30 patients and 20 nurses. Thus, a lot of profile information can be retained without decreasing the performance of the system. Moreover, at least 30 calls can be made simultaneously without influencing the performance.

Third, the portable buttons improve the mobility and the safety of the patients. Patients can walk around freely and make calls. It can be derived from the simulations that it often occurs that patients need to make calls in remote areas such as smoking areas or the restaurant, where there are no nurses present. This problem is of course most prominent in departments where patients are fairly mobile. However, patients could potentially exploit the system as they can call a nurse from anywhere in the hospital even for trivial requests, e.g., a glass of water. This could increase the walking distance and workload of the nurses.

It is however important to note that the developed *oNCS* does not necessarily need to be combined with mobile nurse call buttons to offer advantages. Even when buttons fixed to the walls are used, the *oNCS* offers improvements compared to the traditional nurse call systems as its nurse call algorithm is more dynamic by taking a plethora of context information into account, e.g., risk factors of patients, qualifications of staff and priority of the call. This was already thoroughly addressed in the previous paragraphs.

Finally, the dynamic priority assessment of calls instead of statically defining these priorities leads to a number of advantages. The priority of a call depends on the risk factors of the patients and the kind of call. This means that the priority of a call is adjusted to the specific needs and profile of the patient. This leads to a wider range of priorities of the calls that are made.

The scalability of this probabilistic assessment was presented in Section B.3.3. These results can be improved by calculating the probabilistic values that indicate that the patient is a low, medium or high risk patient during down-time. These are stored as facts in the ontology, significantly reducing the number of probabilistic statements required to determine the priority of a call.

However, our study is limited by the fact that the probabilities in the ontology were only determined by domain experts. These probabilities indicate the probability that a patient belongs to a certain risk group based on the risk factors of this patient. Probabilities were also added to the ontology to express the probability that a call of a particular kind made by a patient from a particular risk group has a particular priority. Collecting data out of which these probabilities could be determined or with which the probabilities could be validated, requires a very extensive study. The study would for example require that nurses note down for each call they handled which priority it had and for which reason the call was made. This requires a significant effort from the already time constrained caregivers. This study was thus not conducted as the goal was to first validate whether incorporating probabilistic priority assessment in the *oNCS* would offer significant benefits. Basing this study on probabilities determined by an expert panel with years of experience in the field gives us a good idea of the impact of the *oNCS*.

However, we do acknowledge that requiring these probabilities as input for the *oNCS* could prove cumbersome to determine or assess by the departments and hospitals where the *oNCS* would be deployed. Therefore, research is on-going on extending the *oNCS* with an autonomic, self-learning component. This component uses data collected about patients and logged by the *oNCS* about its usage to gradually adapt the probabilities such that an optimal configuration of the *oNCS* is obtained for this department or hospital.

B.5 Conclusion

This article described an extension of the Ontology-based nurse call system (*oNCS*) that supports a more sophisticated nurse call algorithm by dynamically assigning priorities to calls based on the risk factors of the patients and the kind of call. The benefits of this novel *oNCS* were illustrated with extensive simulations about data collected from three departments of the Ghent University Hospital. The *oNCS* significantly improves the assignment of nurses to calls. Calls generally have a nurse present faster, the workload-distribution amongst the nurses improves and the priorities and kinds of the calls are taken into account. The execution time of the nurse call algorithm is negligible. Future work will mainly focus on improving the scalability of the probabilistic assessment algorithm to determine the priority of a call.

Acknowledgment

We would like to acknowledge that this research was supported by the head nurses Annemie Danhieux and Johan Van Daele and the assistant head nurse Carine Gevaert. Special thanks also goes out to my colleague Matthias Strobbe for supporting the implementation in the CASP platform. Finally, we would like to acknowledge the people behind *Pronto* and especially Pavel Klinov for always being open to questions and discussions.

Femke Ongenae is funded by the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT).

References

[1] A. Zuker, T. Heart, Y. Parmet, N. Pliskin, and J. S. Pliskin. *Electronic notifications about drug substitutes can change physician prescription habits: a cross-sectional observational study*. Medical Decision Making, 31(3):395–404, 2011.

[2] A. D. Naik and H. Singh. *Electronic health records to coordinate decision making for complex patients: What can we learn from Wiki?* Medical Decision Making, 30(6):722–731, 2010.

[3] E. Bottieau, J. Moreira, J. Clerinx, R. Colebunders, A. V. Gompel, and J. V. den Ende. *Evaluation of the GIDEON expert computer program for the diagnosis of imported febrile illnesses*. Medical Decision Making, 28(3):435–442, 2008.

[4] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, M. V. Gils, A. Jacobs, S. De Zutter, P. Verhoeve, A. Ackaert, and F. De Turck. *Participatory design of a continuous care ontology: towards a user-driven ontology engineering methodology*. In J. Filipe and J. L. G. Dietz, editors, Proc. of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, October 26-29 2011. SciTePress.

[5] F. Ongenae, P. Duysburgh, M. Verstraete, N. Sulmon, L. Bleumers, A. Jacobs, S. D. Z. Ann Ackaert, S. Verstichel, and F. D. Turck. *User-Driven Design of a Context-Aware Application: An Ambient-Intelligent Nurse Call System*. In Proc. of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pages 205–210, San Diego, CA, USA, May 2012. doi:http://dx.doi.org/10.4108/icst.pervasivehealth.2012.24869910.4108/icst.pervasivehealth.2012.248699.

[6] G. W. Wachter. *Hospitals Unplugged: The Wireless Revolution Reaches Healthcare*. Telemedicine Information Exchange, 2001. Available from: http://tie.telemed.org/articles/article.asp?path=articles&article=hospitalsUnplugged_gw_tie01.xml.

[7] E. T. Miller, C. Deets, and R. Miller. *Nurse call and the work environment: lessons learned*. Journal of Nursing Care Quality, 15(3):7–15, 2001.

[8] L. Linden and K. English. *Adjusting the cost-quality equation: Utilizing work sampling and time study data to redesign clinical practice*. Journal of Nursing Care Quality, 8(3):34–42, 1994.

[9] T. R. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199–220, 1993. Available from: http://ontology.csse.uwa.edu.au/reference/browse_paper.php?pid=233281545, doi:10.1.1.101.7493.

[10] F. Ongenae, M. Strobbe, J. Hollez, G. D. Jans, F. D. Turck, T. Dhaene, P. Demeester, and P. Verhoeve. *Design of a semantic person-oriented nurse call management system*. INTERNATIONAL JOURNAL OF WEB AND GRID SERVICES, 4(3):267–283, 2008.

[11] *Ghent University hospital*. http://www.healthcarebelgium.com/index.php?id=uzgent, 2013.

[12] F. Ongenae, D. Myny, T. Dhaene, T. Defloor, D. Van Goubergen, P. Verhoeve, J. Decruyenaere, and F. De Turck. *An ontology-based nurse call management system (oNCS) with probabilistic priority assessment*. BMC Health Services Research, 11:26, 2011.

[13] *Televic NV, specialized in nurse call systems, audio and multimedia communication*. http://www.televic.com, 2013.

[14] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Berlin: Springer Verlag, 2nd edition, 2001.

[15] Y. Yang and J. Calmet. *OntoBayes: An Ontology-Driven Uncertainty Model*. In M. Mohammadian, editor, Proc. of the International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC 2005), pages 457–463, Vienna, Austria, November 28-30 2005. Washington: IEEE Computer Society.

[16] Z. Ding and Y. Peng. *A Probabilistic Extension to Ontology Language OWL*. In Proc. of the 37th annual Hawaii International Conference on System Sciences (HICSS-37), Big Island, Hawaii, January 5-8 2004. Washington: IEEE Computer Society.

[17] K. B. Laskey. *MEBN: A Language for First-Order Bayesian Knowledge Bases*. Artificial Intelligence, 172(2–3):140–178, 2008.

[18] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003. ISBN:0521150116.

[19] T. Lukasiewicz. *Probabilistic Description Logics for the Semantic Web*. Technical report, Technical University of Wien, Institute for Information Systems, Wien, Austria, 2007.

[20] P. Klinov. *Pronto: A Non-monotonic Probabilistic Description Logic Reasoner*. In S. Bechhofer, M. Hauswirth, J. Hoffman, and M. Koubarakis, editors, Proceedings of the 5th European Semantic Web Conference (ESWC), pages 822–826, Tenerife, Canary Islands, Spain, June 1-5 2008. Berlin: Springer. Available from: http://pellet.owldl.com/pronto.

[21] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL reasoner*. Journal of Web Semantics, 5(2):51–53, 2007. Available from: http://pellet.owldl.com/. doi:10.1016/j.websem.2007.03.004.

[22] D. Myny, D. V. Goubergen, V. Limere, M. Gobert, S. Verhaeghe, and T. Defloor. *Determination of standard times of nursing activities based on a Nursing Minimum Dataset*. Journal of Advanced Nursing, 66(1):92–102, 2010.

[23] C. M. Meade, A. L. Bursell, and L. Ketelsen. *Effect of nursing rounds on patients' call light use, satisfaction and safety*. AM J NURS, 106(9):58–70, 2006.

[24] M. Strobbe, G. D. Jans, J. Hollez, N. Goeminne, B. Dhoedt, F. D. Turck, and et al. *Design of an open context-aware platform enabling desk sharing office services*. In H. R. Arabnia, editor, Proc. of the International Conference on Pervasive Systems & Computing (PSC 2006), pages 135–141, Las Vegas, Nevada, USA, June 26-29 2006. CSREA Press.

[25] M. Strobbe, J. Hollez, G. D. Jans, O. V. Laere, J. Nelis, F. D. Turck, B. Dhoedt, P. Demeester, N. Janssens, and T. Pollet. *Design of CASP: an open enabling platform for context aware office and city services*. In T. Pfeifer, J. Strassner, and S. Dobson, editors, Proceedings of the 4th International Workshop on Managing Ubiquitous Communications and Services (MUCS 2007), pages 123–142, Munich, Germany, May 27 2007. Berlin: Multicon Verlag. ISBN:3-930736-07-1. Available from: http://en.scientificcommons.org/23102335.

[26] P. Klinov and B. Parsia. *Optimization and Evaluation of Reasoning in Probabilistic Description Logic: Towards a Systematic Approach*. In A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. F. T, and et al., editors, 7th International Semantic Web Conference (ISWC 2008), pages 213–228. Berlin: Springer, October 26-30 2008.

[27] H. Tzeng, D. L. Ronis, and C. Yin. *Relationship of actual response time to call lights and patient satisfaction at 4 US hospitals*. Journal of Nursing Care Quality, 27(2):E1–E8, 2012.

[28] L. McGillis, C. Pedersen, P. Hubley, E. Ptack, A. Hemingway, C. Watson, and et al. *Interruptions and Pediatric Patient Safety*. Journal of Pediatric Nursing, 25(3):167–175, 2010.

C

Intelligent Task Management Platform for Healthcare Workers

F. Ongenae, T. Vanhove, F. De Backere, and F. De Turck.

Submitted to Journal of Medical Systems, July 2013.

This appendix presents an ontology-based task management application as further demonstration of the general applicability of the developed continuous care ontology, discussed in Chapter 2. This application intelligently assigns priorities and caregivers to tasks based on the continuous care context information caputered in the semantic model.

⋆ ⋆ ⋆

Abstract The medical staff in a hospital could benefit from a specialized task management system, considering their high workload covering different patients. This paper presents an intelligent task management platform that automatically prioritizes and (re-)assigns tasks to the appropriate caregivers based on the current healthcare context captured in a continuous care ontology. Moreover, this platform provides the caregivers with a smartphone allowing them to easily view and process their assigned tasks.

C.1 Introduction

C.1.1 Background

Task management systems have been around for some time and with the growing market share of smartphones and tablets, they have been introduced in everyday life. However, the adoption of such systems within healthcare settings is lagging behind. The medical staff in a hospital could benefit from a specialized task management system, considering their high workload covering different patients.

Time is a valuable resource in healthcare settings and caregivers respond to this time pressure by attempting to work as efficiently as possible by establishing a routine and prioritizing their work tasks [1]. However, each nurse has his or her own way of reaching a routine, e.g., by organizing work per patient or task. Nurses also use different criteria to prioritize tasks, e.g., the complexity of the task or the consequences of a task for the patient, the nurse and other tasks [2]. These strategies also cause nurses to lose the flexibility to respond to events and people as any non-scheduled event is perceived as a disruption or something to be prevented [1]. Continuously monitoring their task load and assessing the priorities of these tasks is a tiresome job. When overloaded with work, nurses also attempt to delegate the work better amongst themselves [3]. However, efficient delegation and re-assignment of tasks is often hindred by the social context. Today, caregivers have to decide for themselves who would be most qualified or able to take over the task and have to locate and contact his person themselves to hand over the task. Assigning tasks to other staff members personally can be a hassle, as different problems can occur. For example, it is not always possible to find or reach a person able to perform the task. Keeping the workload balanced is hard as well if there is no general overview of the division of tasks.

To address these issues, this paper presents an intelligent task management platform that automatically prioritizes and (re-)assigns tasks to the appropriate caregivers based on the current healthcare context, e.g., profile and current condition of the patients, current workload of the caregivers and kind of tasks. Moreover, this platform provides the caregivers with a smartphone allowing them to easily monitor their current workload, automatically re-assign tasks, keep track of tasks that have been assigned to them and order these tasks according to their preference, e.g., by task, by patient or by priority. Moreover, the nurses can indicate which tasks they are currently handling or have handled. Tasks can be created by the caregivers or by another application or service, e.g., decision support tools or regular tasks that can be created automatically.

C.1.2 Objectives

The aim of this research is the design of a continuous care task management platform with automatic priority and task assignment. The application should offer the advanced features listed below:

- ***Delegation***: A caregiver can assign a staff member to a task, when the task is created. However, it must also be possible to automatically assign the most appropriate caregiver(s) to tasks based on the current context. Similarly, when a task is re-assigned, the caregiver can be assigned by the staff member of automatically by the system.
- ***Dynamic priority assignment***: A priority can be assigned to the task when it is created by the healthcare worker. However, the platform should also be able to assign or change the priority of the call based on the current context.
- ***Performance***: The smartphone application should be able to give an overview of all the tasks assigned to a specific person within a second. This means that the priorities assigned to these tasks should also be calculated with negligible delay. This list should also be updated correctly and timely to alert the user of new tasks that have been assigned.
- ***Scalability***: The platform should be able to support a lot of simultaneous users. An increasing amount of users may not cause a bottleneck for gathering the tasks assigned to a specific person and determining their priority.
- ***User-friendly***: The graphical user interface (GUI) of the task management application on the smartphone must be easy and intuitive to ease the integration of the application in the day-to-day work of the users. An intuitive overview of the different tasks assigned to a healthcare worker and their status is needed which can be easily searched and ordened. A user should also be able to quickly indicate that he or she has started or finished a task. Adding new tasks should require a minimum of input as text-based data entry on a smartphone is tedious. Finally, special alerts should be generated when tasks have been assigned with a high priority.
- ***Reliable***: Three kinds of faults can occur: the back-end server/database can go down, task information is not delivered to the smartphone of the staff member or the information of the smartphone is not delivered to the back-end. The platform has to be able to cope with each of these situations.
- ***Adaptability***: The application has to cope with internal and external changes. The user should be able to change the methods the system uses to assign or change priorities of tasks. The administrators should also be able to update or tests parts of the platform, e.g. the database, without downtime.

- ***Security***: Security is important to this platform as medical data from patients is accessed. First, violations on the inside muste be avoided. It is not allowed that any nurse or doctor has access to tasks that are not assigned to him or her or the role that this person belongs to. Additionally, the priority assignment Rules and settings of the platform should only be changed by persons with the correct authorization. Second, the system must also ensure authentication, so that violations from the outside are impossible.

C.1.3 Paper organization

The remainder of this paper is organized as follows. Section C.1.4 details the architecture of the developed task management system. The continuous care ontology, which is used to capture the current healthcare context, is discussed in Section C.1.5, while Sectionr̃efsec:taskMgmt:algo dives deeper into the algorithm to assign the most appropriate caregiver to a task based on the current context captured in this ontology. Section C.1.7 elaborates on the implementation details and the evaluation set-up. The results are discussed in Section C.1.8. Finally, Section C.2 highlights the conclusions.

C.1.4 Architecture

Figure C.1 visualizes the high-level architecture of the task management system. It consists of a combination of 2 architectureal design patterns [4], namely *Publisher-Subscriber* and *Model-View-Controller*. The first is a messaging pattern, which allows that components, called *Publishers*, send a message to a *Message Handler*. Each message is associated with a type. *Subscribers* can then subscribe themselves through the *Message Handler* to specific types of messages. The publisher is thus unaware of which components, if any, receive the message. This leads to a very scalable and dynamic architecture in which the components are loosely coupled. The second pattern separates the representation of information from the users' interaction with it. The *Model* contains the logic and the data of the system, while the *View* is responsible for representing the data to the users. Because of this seperation of concerns, different representations of the data can easily be implemented. The *Controller* is responsible for communicating the actions of the users on the *View* to the *Model*. In case the data in the *Model* changes, the *Views* are automatically updated.

In this case, the *Publishers* can either be the smartphones of the caregivers or various healthcare services, e.g., decision support tools or a work schedule application. These *Publishers* publish messages to the *CommunicationHandler*. The messages can for example be new tasks that are created or reassigned, staff members logging in or out or task information that is updated, e.g., the status changes to finished or its priority is altered. *Subscribers*, i.e., the task applications running on

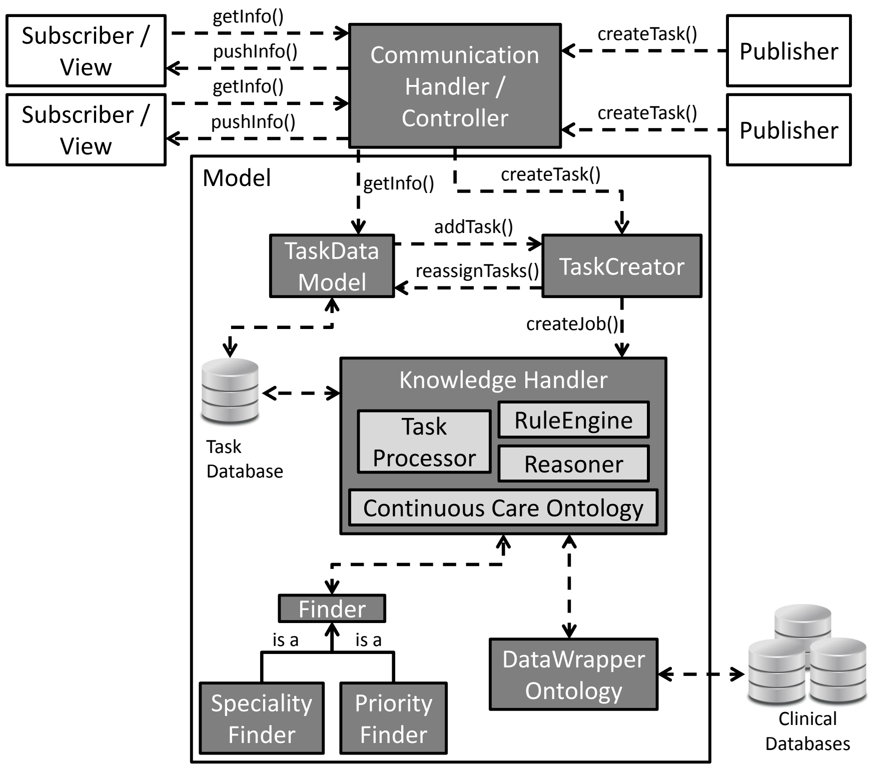


Figure C.1: High-level architecture of the task management system

the smartphones, which also play the role of the *Views*, can then subscribe to the messages they are interested in. These messages are automatically pushed to these *Subscribers*. However, based on interaction of users with the *View* is is also possible that data is requested from the *CommunicationHandler*. This push/pull-model increases the flexibility of the system.

The *Model* consists of various components, which are able to process the received messages of the *Publishers*. The *CommunicationHandler* is responsible for all the communication between the *Publishers* and the *Subscribers*. The *TaskDataModel* component is responsible for managing all the data about the current tasks. Past tasks are stored in the *Database*. This *Database* is also used as back-up to cope with sudden failures of the system. The *TaskCreator* is responsible for creating new tasks based on the information received from the *Publishers*. If not all information is provided, e.g., the person who should handle the task or its priority, this information is requested from the *KnowledgeHandler*. When the task is created, it is passed to the *TaskDataModel*.

The *KnowledgeHandler* can again be broken down into different components. This component manages all the knowledge about the current context in an ontology, e.g., risk factors and medical information of patients and compentences, roles, locations and tasks of the caregivers. An ontology [5] is a formal and semantic model of all the concepts within a particular domain and their relationships and properties. This commen data-format can then be used to integrate all the healthcare data in a formal manner. The data, which adheres to the concept definitions in the ontology, is collected by the *DataWrapperOntology* from the *Clinical Databases* present in a healthcare institution, e.g., Electronic Patient Records (EPR), Laboratory databases or a database containing personnel information. A semantic *Reasoner* is a piece of generic software, which is able to infer logical consequences, i.e., new knowledge, out of the information captured in an ontology. For example, it can be used to determine which staff members have the appropriate competences to execute a certain task. More complex reasoning on data can be performed by a *Rule Engine*.

The *TaskProcessor* is responsible for processing the tasks it receives from the *TaskCreator*. It contains a queue to be able to process the different requests one by one in the correct order. The *TaskProcessor* is responsible for gathering the different information needed to process the task from the *Rule Engine* and the Ontology *Reasoner*. It then passes this information to the *Finders*, which implement the algorithms to assign the most appropriate priority (*PriorityFinder*) and caregiver (*SpecialityFinder*) to the task based on the gathered information.

C.1.5 Continuous care ontology

A modular continuous care ontology was developed, modeling context information and knowledge utilized across the various continuous care settings, i.e., hospitals, residential care and homecare. It consists of a high-level ontology and two low-level ontologies. The first, called the continuous care core ontology, contains knowledge that is applicable across all continuous care domains and is of interest to a plethora of healthcare applications and services. The core ontology was designed in a modular way instead of as one big semantic model, which facilitates (partial) re-use. The following seven core ontologies were developed: the *Upper*, *Sensor*, *Context*, *Profile*, *Role* & *Competence*, *Medical* and *Task* continuous care core ontologies. Two low-level ontologies were developed, modeling knowledge particular to a specific continuous care domain, namely the low-level *Cure* and *Care* ontology, which are respectively tuned towards the knowledge exchanged in hospital and continuous care settings. Each of these models also consists of a number of ontologies, extending specific core ontologies. More information about these ontologies can be found in Ongenae, et al. [6].

The prevalent concepts of the continuous care ontology for the task manage-

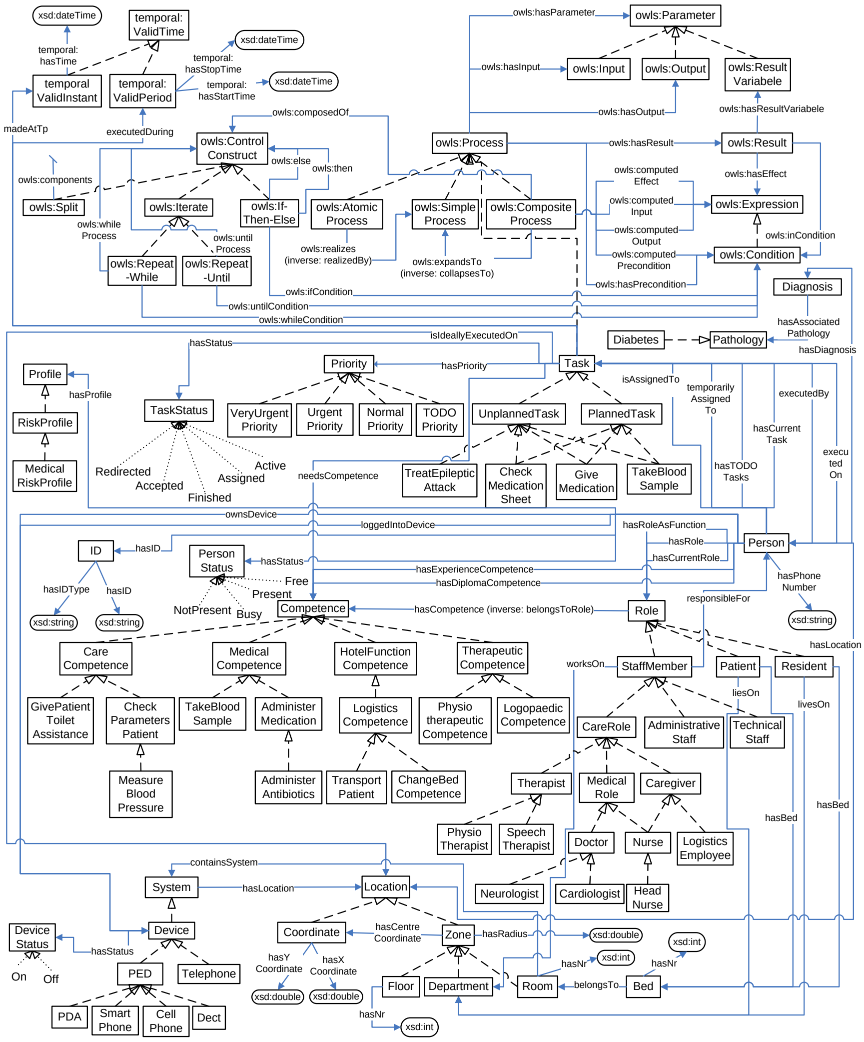


Figure C.2: Overview of the most prevalent classes and relations of the continuous care ontology for the task management platform

ment platform are visualized in Figure C.2. The people present in the healthcare environment are represent by the `Person` concept. Each person is asssociated with a telephone number. For the staff members, this is the phone number of their smartphone, which runs the mobile task application. People can also be associated with the devices they own and are logged into through the `loggedIntoDevice` and `ownsDevice` relations. This is used to know to which devices the notifications about the assigned tasks should be sent. The `Location` of a person is indicated, which can either be a `Coordinate` or a `Zone`. Each person is unique identified by his or her `ID`. The status of the person can also be expressed, e.g., `Busy`, `Free`, `Present` or not. To express the capabilities of the people, each `Person` is associated with one or more `Roles`. Each `Role` is defined by its `Competences` through classification axioms, e.g., the `Doctor` is defined as a role which has all the `MedicalCompetences`. Each person is associated with competences and roles through five relationships. The `hasFunction` relation indicates the primary role of a person, while `hasRole` models all the roles this person can have and the `hasCurrentRole` models his or her current role. The `hasDiplomaCompetence` and `hasExperienceCompetence` relationship model all the competences this person has acquired, either through their diploma and following courses or by experience. Some example roles and competences are shown in Figure C.2. Finally, the medical information about patient is expressed using the `hasDiagnosis` relationship. Based on the medical information, the `MedicalRiskProfile` of a patient is determined.

To represent the tasks and continuous care workflows processes, the *OWL-S Process* [7] ontology was imported, of which the classes are preceded by the `owls` namespace prefix in Figure C.2. The `Process` concept models a process, which can return information and produce a change in environment based on the context and the information it is given. This is described by `hasInput`, `hasOutput`, `hasPrecondition` and `hasEffect` relations. A process can be composed of several other processes. How these processes are combined is expressed by the `ControlConstruct` concept. The `Task` concept, introduced as subclass of `Process`, represents the various continuous care tasks. It is further divided into planned and unplanned tasks. Each task has also an associated `Status`, e.g., `Assigned` or `Finished`, and `Priority`. Each task is defined by the `Competences` which are needed to execute this task. Also the location at which this task is preferably executed can be indicated. Finally, the time is indicated at which the task was created and when it was executed using concepts of the *SWRLTemporalOntology* [8]. Some example tasks are visualized in Figure C.2.

C.1.6 Task assignment algorithm

The task management system uses the information captured in the continuous care ontology to assign the most appropriate caregiver to a task. The task assignment algorithm consists of four steps, namely assessing the priority of the task, determining the competences needed to execute the task, filtering the qualified possible caregivers and choosing one of these candidates to assign the task to.

When a task is created by a caregiver on the mobile application or by another service, the following information is specified: the name of the task, its priority and category and the associated patient. The *Rule Engine* contains rules, which are able to determine the competences needed to execute determine based on the specified category of the task. For example, the following rule specifies that a task of the `Medication` category requires a caregiver with the `AdministerMedication` and `CheckMedicationSheet` competences:

```
rule ``Medication"
      when
           $t :  Task (category == ``Medication")
      then
           ($t).setCompetency(``CheckMedicationSheet")
           ($t).setCompetency(``AdministerMedication");
end
```

A `Task` of the correct category is created in the ontology with the indicated name and the needed competence is specified using the `needsCompetence` relationship. The associated patient is modeled using the `executedOn` relationship, while the priority is associated using the `hasPriority` relation. Next, the following two rules are specified in the ontology, which allow the *Reasoner* to determine the caregivers, who are able to execute the task because they have the competences required for the task and they are currently present in the healthcare setting:

```
Person(?p), Competence(?c), Task(?t),
hasDiplomaCompetence(?p, ?c), needsCompetence(?t, ?c),
hasStatus(?p, Present)
-> temporarilyAssignedTo(?t, ?p)

Person(?p), Competence(?c), Task(?t),
hasExperienceCompetence(?p, ?c),
needsCompetence(?t, ?c), hasStatus(?p, Present)
-> temporarilyAssignedTo(?t, ?p)
```

The caregivers, who fullfill these criteria, are temporarily assigned to the task.

Next, the priority of the task is assessed. The ontology contains rules, which specify whether the patient has a `MedicalRiskProfile` based on the medical information captured about this patient in the ontology. For example, when the patient has been transferred from the ICU in the last 72 hours or when he or she has recently had a heart attack or an epileptic seizure, he or she is considered at risk. When the patient has a medical risk, his or her priority is increased one category. This algorithm is implemented in the *PriorityFinder*, which interacts with the information in the continuous care ontology.

Out of all these temporarily assigned caregivers, the most appropriate one is chosen using a weighted algorithm implemented in the *SpecialityFinder*, which

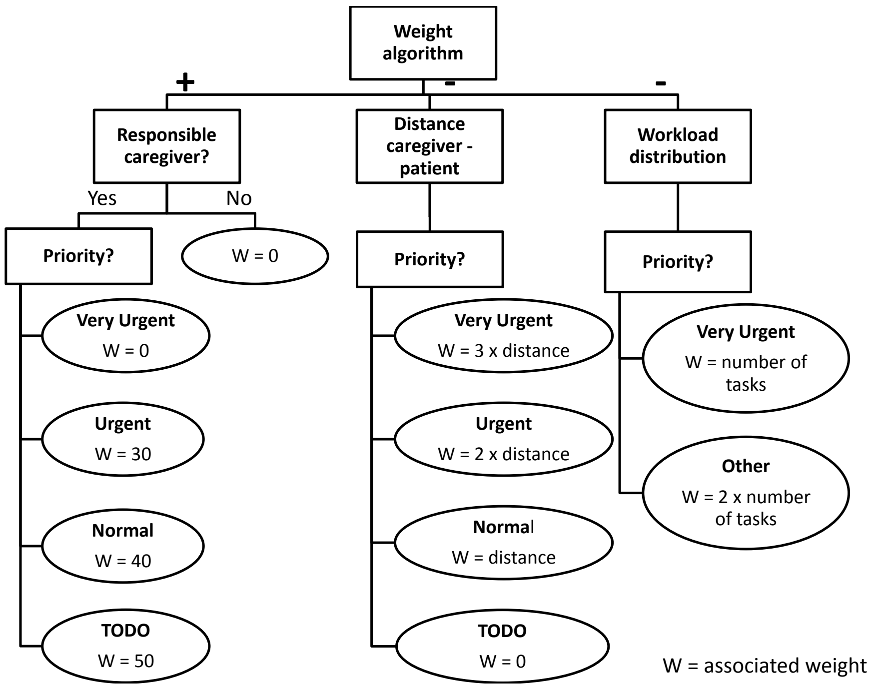


Figure C.3: Algorithm to assign weights to the qualified caregivers to choose the most appropriate one to handle a task

takes into account the context information captured in the ontology. Using the algorithm visualized in Figure C.3, each temporarily assigned caregiver is given a weight. The algorithm takes three factors into account, namely the distance between the patient and the caregiver, the relationship between the patient and the caregiver and the workload distribution. The priority of a task determines how much these factors are taken into account. The relationship between caregivers and patients is taken more into account for lower priority tasks. A caregiver, who is more familiar with the medical situation of a patient will be able to perform the task more quickly and easily. Moreover, it gives the patient a feeling of security, trust and continuity when the same caregiver performs most tasks. The weight associated with this factor is first calculated. The distance between the caregiver and the patient becomes more crucial for higher priority tasks. The further the caregiver is removed from the patient, the longer it will take before he or she is able to handle the urgent task. Depending on the priority, the distance is substracted three or less times from the weight, which was already calculated for this caregiver. Finally, it is important to evenly distribute the workload across the various caregivers. Based on the priority of the task, the number of tasks already assigned to the caregiver are substracted two or one time from the already calculated weight.

For high priority calls, the workload distribution is taken less into account. This factor is less important than the other two factors, as caregivers, who are responsible for more patients, will naturally have more tasks assigned to them. It is mostly used to choose between two or more caregivers, who received an equal weight based on the previous two factors. Finally, the caregiver who received the highest weight is chosen.

The chosen caregiver is assigned to the task in the ontology using the `isAssignedTo` relationship. The `temporarilyAssignedTo` relationships are not removed from the ontology so that they can be reused when the task needs to be re-assigned, e.g., because the staff member logs out or because he or she is too busy or unable to handle the task. The caregiver can indicate that the task should be re-assigned on the smartphone.

C.1.7 Implementation details and evaluation set-up

The continuous care ontology was implemented in the Web Ontology Language (OWL) [9] using the Protégé [10] ontology editor. The rules in the ontology were expressed using the Semantic Web Rule Language (SWRL) [11]. Drools [12] was used as Rule Engine. Both Pellet [13] and Hermit [14] were employed as ontology reasoners to evaluate which has the best performance. PubNub [15] handles the communication between the smartphone and the back-end server. The mobile application was implemented in Android 2.3.

It is important to evaluate the performance, i.e., execution time and memory usage, of the developed task management system. Most healthcare environments have a limited amount of resources and delegating the processing to the cloud is often difficult because of privacy issues. Moreover, it is important that tasks are swiftly delegated, so that urgent tasks can quickly be assessed and handled. To evaluate the performance, each test consists of the creation of a task on the task application and using the task management system to assign the most appropriate priority and caregiver. To evaluate the execution time and memory usage, the parameters of the system, e.g., the amount of caregivers in the ontology or the amount of rules, is gradually increased.

To achieve reliable results, each test was repeated 35 times, of which the first three and the last two were omitted during processing. Finally, the averages across the 30 remaining runs are calculated and visualized in the form of graphs. The tests were performed on a Acer Aspire 5920 with the following specifications: 4 gigabyte (GB) 800 megahertz (MHz) Double Data Rate Synchronous Dynamic Random-Access Memory (DDR2 SDRAM) and an Intel Core 2 Duo T5550 1.83 gigahertz (GHz) Central Processing Unit (CPU).

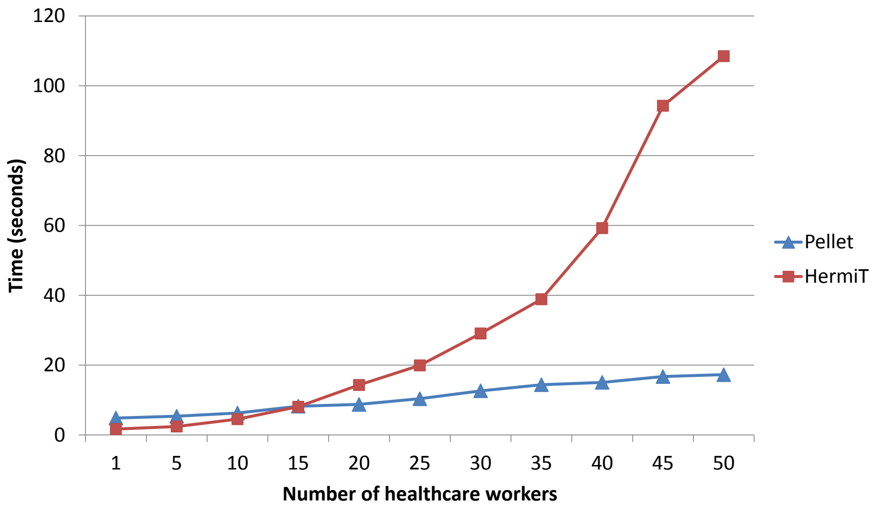
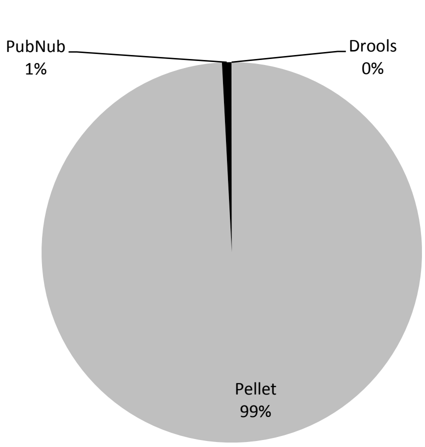


Figure C.4: Execution time of the task management system as a function of the amount of present healthcare workers

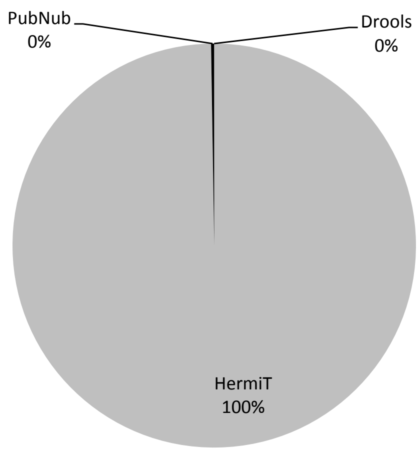
C.1.8 Results & discussion

Figure C.4 shows the execution time of the task management system when the amount of available healthcare workers available in the ontology is gradually increased. Each healthcare worker also has the competence that is needed to execute the task that is being assigned during the test. Consequently, each present caregiver can possibly be assigned to the task. As mentioned previously, the task management system can be implemented using different semantic reasoners. The graph shows the execution time of the entire system when Pellet or Hermit are used as semantic reasoners. It can be noted that the implementation with Hermit performs significantly worse than with Pellet when more than 15 healthcare workers are present in the ontology. The performance of the implementation with Pellet stays belong 20 seconds, which is acceptable. The Pellet implementation has a linear trend, while the Hermit implementation has an exponential one. When 50 caregivers are present, the execution time of the Hermit implementation is around 2 minutes, which is unacceptable.

To analyze this result in more depth, Figure C.5 shows the average distribution of the execution time across the three main components of the task management system, namely the communication using PubNub, the rule reasoning using Drools and the semantic reasoning using Pellet or Hermit. As PubNub is a cloud-based solution, it was difficult to accurately measure the execution time. However, PubNub publishes data on its website about the current response time and cloud uptime. At the time of the research the response time was on average 177 milliseconds (ms).



(a) Task management system implemented with the Pellet semantic reasoner



(b) Task management system implemented with the Hermit semantic reasoner

Figure C.5: Distribution of the total execution time of the task management system across the three main components: PubNub, Drools rule engine and the semantic reasoner

Methods	**Pellet**	**Hermit**
	execution time (seconds)	
getPossibleCandidates	16.98	0.43
getLocation	0	61.12
isResponsibleCaregiver	0	0
workloadPerson	0	0

Table C.1: Average execution time of the different methods requiring semantic reasoning

The test was run with 15 caregivers and 23 patients, which is a realistic dataset based on data gathered about departments at Ghent University Hospital. 1000 Drools rules were present. As expected, the semantic reasoning consumes almost all the execution time.

To identify the bottlenecks in the semantic reasoning, Table C.1 shows the average execution time of the different methods that require semantic reasoning. The same ontology is used as in the previous test. The `getPossibleCandidates` method retrieves from the ontology the caregivers with the appropriate competences to execute the task that is currently being assigned. The `getLocation` method retrieves the location of a particular person. This information is used to calculate the distance between a caregiver and a patient. The `isResponsibleCaregiver` method retrieves whether a caregiver is responsible for a particular patient. The `workloadPerson` method retrieves the number of tasks currently assigned to a person. The latter three methods are used by the weighted algorithm,

Methods	Pellet execution time (seconds)
getLocation	15.76
getPossibleCandidates	0
isResponsibleCaregiver	0
workloadPerson	0

Table C.2: Average execution time of the different methods requiring semantic reasoning with Pellet when the `getLocation` method is called before the `getPossibleCandidates` method

Methods	Hermit execution time (seconds)
getObjectPropertyValues	0.18
isIndividualofClass	0
getDataPropertyValues	60.93
getLiteral	0s

Table C.3: Average execution time of the different steps of the `getLocation` method when Hermit is used as semantic reasoner

which assigns the most appropriate caregiver to the task.

It can be noted that for the Pellet implementation the `getPossibleCandidates` requires the most execution time. This result is however misleading. This is the first semantic reasoning task performed during a test. At this point, Pellet will check the consistency of the ontology and classify it for the first time. Consequently, this method requires a lot of time. As no information is added to the ontology before the next reasoning requests, these queries are performed on the already classified ontology. The following reasoning tasks are thus performed very swiftly. This is demonstrated by the results in Table C.2 where the order of the reasoning methods in the implementation was changed. It can be noted that the `getLocation` method, which is now the first reasoning task, requires the most execution time.

For the Hermit implementation, the `getLocation` method requires the most time. As this is not the first reasoning method, it is studied further to analyze its bottleneck. Table C.3 shows the execution time of the different steps of the `getLocation` method. The `getObjectPropertyValues` retrieves the location of the person from the ontology. The `isIndividualOfClass` method checks whether this location is a `Zone` or a `Coordinate` and is responsible for mapping the `Zone` to its `Coordinate(s)`. The `getDataPropertyValues` method retrieves actual X and Y coordinates from the ontology, while the `getLiteral` method retrieves the double value from these coordinates. It can be noted that the `getDataPropertyValues` requires the most amount of time.

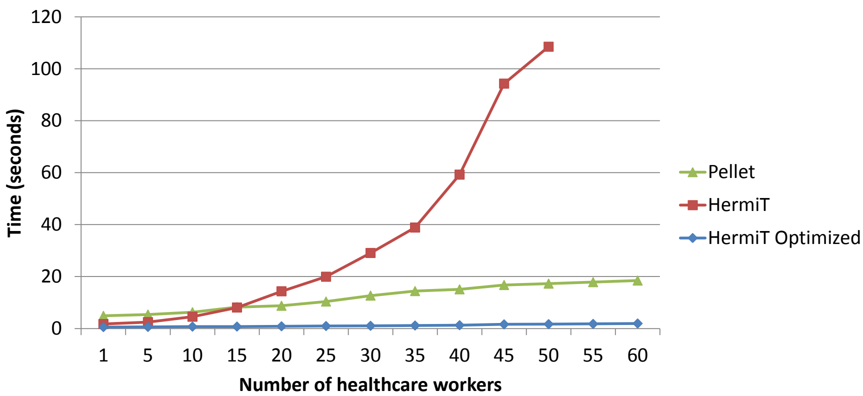


Figure C.6: Execution time of the task management system as a function of the amount of present healthcare workers for the three possible implementations of the semantic reasoning

To resolve this issue, the implementation of the `getLocation` method was optimized, so that the datatype properties, namely `hasXCoordinate` and `hasX-Coordinate`, do not need to be retrieved. To achieve this, the name of the `Co-ordinate` was formatted as follows: $< X\text{−}coordinate >:< Y\text{−}coordinate >$. For example, a coordinate with $X = 50$ and $Y = 50$ receives the name "50 : 50". Now only the two first steps of the getLocation method, namely `getObject-PropertyValues` and `isIndividualofClass`, need to be executed. The name of the `Coordinate`, which is returned as a result of these two steps, can then be analyzed to retrieve the X and Y coordinate. Figure C.6 shows the execution time of the task management system as a function of the amount of healthcare workers for the three possible implementations of the semantic reasoning, i.e., with Pellet, Hermit and with the optimized implementation of `getLocation` for Hermit. It can be noted that the latter has the best performance. The execution time stays below 5 seconds when at most 60 healthcare workers are represented in the ontology.

C.2 Conclusion

In this paper an intelligent task management platform was presented that automatically prioritizes and (re-)assigns tasks to the appropriate caregivers based on the current healthcare context captured in a continuous care ontology. Moreover, this platform provides the caregivers with a smartphone allowing them to easily monitor their current workload, automatically re-assign tasks, keep track of tasks that have been assigned to them and order these tasks according to their preference. Finally, the performance of the system was studied in depth. A task can be assigned in less than 5 seconds when at most 60 healthcare workers are managed by the system. Future work will mainly focus on more intelligent algorithms to assign priorities to the tasks.

Acknowledgment

Femke Ongenae would like to thank the IWT, Institute for the promotion of Innovation through Science and Technology in Flanders, for supporting this work through her PhD grant.

References

[1] B. J. Bowers, C. Lauring, and N. Jacobson. *How nurses manage time and work in long-term care*. Journal of Advanced Nursing, 33(4):484–491, 2001.

[2] C. Hendry and A. Walker. *Priority setting in clinical nursing practice: literature review*. Journal of Advanced Nursing, 47(4):427–436, 2004.

[3] S. Waterworth. *Time management strategies in nursing practice*. Journal of Advanced Nursing, 43(5):432–440, 2003.

[4] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 2nd edition, 2003.

[5] T. R. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199–220, 1993.

[6] F. Ongenae, L. Bleumers, N. Sulmon, M. Verstraete, M. V. Gils, A. Jacobs, S. De Zutter, P. Verhoeve, A. Ackaert, and F. De Turck. *Participatory design of a continuous care ontology: towards a user-driven ontology engineering methodology*. In J. Filipe and J. L. G. Dietz, editors, Proc. of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pages 81–90, Paris, France, October 26-29 2011. SciTePress.

[7] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. *OWL-S: Semantic Markup for Web Services*. Technical report, W3C Member Submission, 2004. Available at: http://www.w3.org/Submission/OWL-S/.

[8] M. J. O'Connor and A. K. Das. *A lightweight model for representing and reasoning with temporal information in biomedical ontologies*. In Proc. of the International Conference on Health Informatics (HEALTHINF), pages 90–97, Valencia, Spain, 2010.

[9] D. L. McGuinnes and F. Van Harmelen. *OWL Web Ontology Language overview*. Technical report, W3C Recommendation, 2004. Available at: http://www.w3.org/TR/2004/REC-owl-features-20040210.

[10] T. H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. In Proc. of the 3rd International Semantic Web Conference, pages 229–243, Hiroshima, Japan, 2004. Available at: http://protege.stanford.edu/.

[11] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Technical report, W3C Member Submission, 2004. Available at: http://www.w3.org/Submission/SWRL/.

[12] M. Bali. *Drools JBoss Rules 5.0 Developer's Guide*. Packt Publishing, Birmingham, UK, 2009.

[13] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. *Pellet: A practical OWL-DL Reasoner*. J Web Semant, 5(2):51–53, 2007. Available at: http://pellet.owldl.com/.

[14] B. Motik, R. Shearer, and I. Horrocks. *Hypertableau Reasoning for Description Logics*. Journal of Artificial Intelligence Research, 36:165–228, 2009.

[15] PubNub. http://www.pubnub.com/.

D

Time Series Classification for the Prediction of Dialysis in Critically Ill Patients using Echo State Networks

F. Ongenae, S. Van Looy, D. Verstraeten, T. Verplancke, D. Benoit, F. De Turck, T. Dhaene, B. Schrauwen, J. Decruyenaere.

Published in Engineering Applications of Artificial Intelligence, Volume 26, Issue 3, Pages 984-996, March 2013.

This appendix thoroughly evaluates the advantages of using Echo State Networks (ESNs) to classify medical time series instead of other traditional classifiers combined with feature extraction and selection. Echo state networks can be integrated in the self-learning framework to discover new knowledge in time-dependent data, as thoroughly discussed in Chapter 7.

⋆ ⋆ ⋆

Abstract *Objective:* Time series often appear in medical databases, but only few machine learning methods exist that process this kind of data properly. Most modeling techniques have been designed with a static data model in mind and are not suitable for coping with the dynamic nature of time series. Recurrent Neural Networks (RNN) are often used to process time series, but only a few training algorithms exist for RNNs which are complex and often yield poor results.

Therefore, researchers often turn to traditional machine learning approaches, such as support vector machines (SVM), which can easily be set up and trained and combine them with feature extraction (FE) and selection (FS) to process the high-dimensional temporal data. Recently, a new approach, called echo state networks (ESN), has been developed to simplify the training process of RNNs. This approach allows modeling the dynamics of a system based on time series data in a straightforward way.

The objective of this study is to explore the advantages of using ESN instead of other traditional classifiers combined with FE and FS in classification problems in the intensive care unit (ICU) when the input data consists of time series. While ESNs have mostly been used to predict the future course of a time series, we use the ESN model for classification instead. Although time series often appear in medical data, little medical applications of ESNs have been studied yet.

Methods and material: ESN is used to predict the need for dialysis between the fifth and tenth day after admission in the ICU. The input time series consist of measured diuresis and creatinine values during the first 3 days after admission. Data about 830 patients was used for the study, of which 82 needed dialysis between the fifth and tenth day after admission. ESN is compared to traditional classifiers, a sophisticated and a simple one, namely support vector machines and the naive Bayes (NB) classifier. Prior to the use of the SVM and NB classifier, FE and FS is required to reduce the number of input features and thus alleviate the curse dimensionality. Extensive feature extraction was applied to capture both the overall properties of the time series and the correlation between the different measurements in the times series. The feature selection method consists of a greedy hybrid filter-wrapper method using a NB classifier, which selects in each iteration the feature that improves prediction the best and shows little multicollinearity with the already selected set. Least squares regression with noise was used to train the linear readout function of the ESN to mitigate sensitivity to noise and overfitting. Fisher labeling was used to deal with the unbalanced data set. Parameter sweeps were performed to determine the optimal parameter values for the different classifiers. The area under the curve (AUC) and maximum balanced accuracy are used as performance measures. The required execution time was also measured.

Results: The classification performance of the ESN shows significant difference at the 5% level compared to the performance of the SVM or the NB classifier combined with FE and FS. The NB + FE + FS, with an average AUC of 0.874, has the best classification performance. This classifier is followed by the ESN, which has an average AUC of 0.849. The SVM + FE + FS has the worst performance with an average AUC of 0.838. The computation time needed to pre-process the data and to train and test the classifier is significantly less for the ESN compared to the SVM and NB.

Conclusion: It can be concluded that the use of ESN has an added value in predicting the need for dialysis through the analysis of time series data. The ESN requires significantly less processing time, needs no domain knowledge, is easy to implement, and can be configured using rules of thumb.

D.1 Introduction

Time series are a special kind of input data to machine learning problems. Most modeling techniques have been designed with a static data model in mind and are not suitable for coping with the dynamic nature of time series. Most dynamic data models are very complex in both design and training algorithms. Examples of such models based on artificial neural networks are the hidden control neural network [1], the neural prediction model [2], the linked predictive neural network [3] and the adaptive time-delay neural network [4]. Recurrent Neural Networks (RNNS) are often used [5] since this type of artificial neural network can represent high-dimensional nonlinear temporal data. Hidden Markov models [6] and neural network - hidden Markov model hybrids [7, 8] are also used to model time series data. An obstacle when using RNNS is that only a few training algorithms exist which are complex and often yield poor results [9, 10].

More recently, three approaches to simplify the training process of RNNS were independently developed. These approaches are liquid state machines (LSM) [11], echo state networks (ESN) [12], and backpropagation decorrelation (BPDC) [13]. The underlying idea of these three methods is similar and nowadays they are referred to as *reservoir computing* [14]. Reservoir computing has become a vivid research field and recently a special issue of "*Neural Networks*" was dedicated to it [15].

The key idea in reservoir computing is that the dynamic system producing the time series data is modeled in a *reservoir* consisting of a RNN. The reservoir is then read by a linear readout function, which is illustrated in Figure D.1. The output of this readout function can then be used to make several kinds of predictions. The training algorithm only affects the linear readout function. For training linear functions many algorithms exist such as linear regression [16].

The goal of this study is to verify whether the use of reservoir computing methods is an added value in classification problems in the intensive care unit (ICU) when the input data consists of time series. We select a case study that is easily characterized by medical experts. This medical classification problem is then handled using reservoir computing, which can directly cope with time series data, and the performance is compared to more traditional machine learning approaches, which cannot directly cope with this high-dimensional temporal data and thus need to be combined with feature extraction (FE) and selection (FS) to process the time series.

LSMs and ESNs are the two pioneering reservoir computing methods. How-

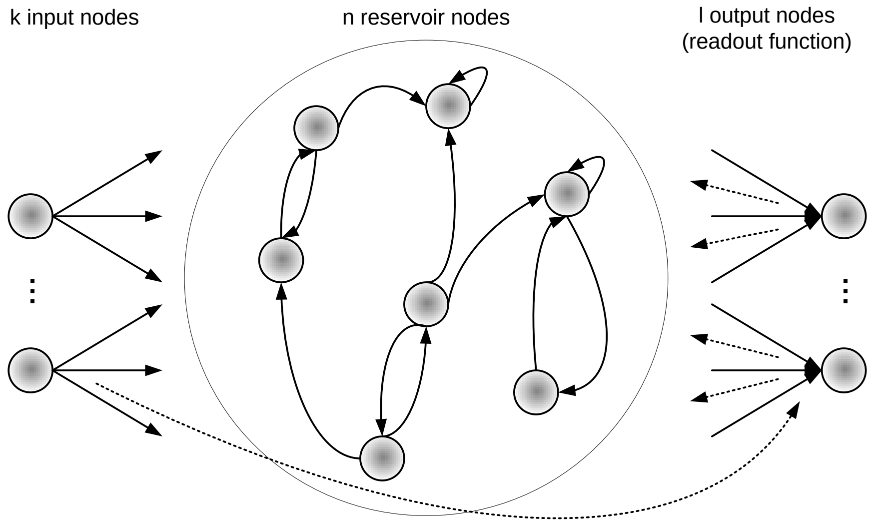


*Figure D.1: **The general layout of an echo state network.** Circles represent input, reservoir, and output nodes. Arrows represent non-zero weighted connections. Dotted arrows denote optional connections.*

ever, the two methods have a very different background [15]. The initial ESN publications were framed in settings of machine learning and nonlinear signal processing applications [10, 12, 17–19]. In contrast, LSMs were developed from a computational neuroscience background, aiming at elucidating principal computational properties of microcircuits [11, 20–22].

This difference in background also explains the main difference between LSMs and ESNs [23]. ESNs standardly use simple sigmoid neurons or leaky integrator neuron models, while LSMs use more sophisticated and biologically realistic models built from a spiking neuron model called the Leaky Integrate and Fire (LIF) neuron [24] and dynamic synaptic connection models [25] in the reservoir. Since the model of both the connections and the neurons themselves in LSMs is quite sophisticated, it has a large number of free parameters to be set, which is done manually, guided by biologically observed parameter ranges. The parameters of ESNs, e.g., the warm-up drop and the leak rate, are more intuitive and can easily be set by using rules of thumb or performing parameter sweeps. Moreover, LSMs require pulse trains as input data. Translating continuous data, of which the training data of the medical problem under study in this research consists, to pulse trains is a complex problem. Consequently, LSMs are usually more difficult to implement, to correctly set up and tune, and typically more expensive to emulate on digital computers than simple ESN-type "weighted sum and non-linearity" RNNs. Thus LSMs are less widespread for engineering applications of RNNs than ESNs. This makes ESNs the better choice for "simple" engineering tasks, such as the medical classification problem under study in this research.

The idea of separation between a reservoir and a readout function has also been arrived at from the point of view of optimizing the performance of the RNN training algorithms that use error backpropagation. It was found that the Atiya-Parlos recurrent learning (APRL) rule [26] restricts the adaptation of the weights to the output layer, effectively splitting the RNN into a reservoir and a readout layer. The outputs weights are trained and the internal weights are only globally scaled up or down a bit [27]. This lead to a learning rule for RNNs called BPDC. Here too, sigmoidal neurons are used, but a significant difference between BPDC reservoirs and ESNs is the fact that feedback connections from the readout layer into the reservoir and into the readout layer itself are used, whereas in practice this is hardly ever the case for ESNs [14]. As for the medical classification task under scrutiny these feedback connections are not needed, ESNs were used instead of BPDC in this research.

More information about the different reservoir computing methods and their various properties and application domains can be found in Verstraeten et al., Jaeger et al., and Lukoševičius and Jaeger [14, 15, 23].

Thus, the ESN was selected as reservoir computing method to handle the medical classification problem studied in this research. The medical time series are also classified using support vector machines (SVM) and the naive Bayes (NB) classifier. This way, we can compare the performance of two traditional classifiers - a sophisticated and a simple one - and the recent classifier based on ESN.

Although medical data are often time series, little medical applications of ESN have been studied yet. To our knowledge, apart from this study, of which a preliminary report has been published which focusses on the clinical aspect of the study [28], ESN have been applied to two other medical use cases. An abstract reported the classification of autistic and normal children [29] and a study described the detection of epileptic seizures on rat data using reservoir computing [30, 31].

In time-oriented medical studies, longitudinal data analysis is a popular approach. However, this is only suitable for relatively short time series - typically up to 10 measurements per input parameter - since longitudinal data analysis focuses on the correlation of measurements within a time series, which diminishes when the time series grows and measurements lie further apart in time [32]. Another approach is repeatedly performing data analysis only in a very small interval or individual points in time. However, this neglects the temporal nature of the data almost completely.

The remainder of this paper is structured as follows. The application data is described in Section D.2. In Section D.3 the classification, feature extraction and selection, and performance evaluation methods used in this study are briefly introduced. Section D.4 then summarizes the experimental setup, after which the results are presented in Section D.5. These are discussed in Section D.6 after which a conclusion is formulated in Section D.7.

D.2 Application data

Since we want to explore the advantages of the use of echo state networks in this study, a simple problem is selected. That is, a problem that is easily solved by an expert in the field. This way, we are sure that the required information to solve the problem is contained in the data and that the acquired result is the outcome of the used method, not the used data.

In collaboration with the ICU department of the Ghent University we selected the problem of predicting whether or not a patient will need dialysis between five and ten days after admission in the ICU. The prediction is made at hour 72 after submission, so only the diuresis and creatinine values of the first three days after ICU admission were retrieved from the ICU database for each patient included in the study. The study population consisted of an observational cohort of 916 patients admitted consecutively to the ICU between May 31st 2003 and November 17th 2007. These patients were selected from a total of 9752 medical and surgical ICU (MICU/SICU) patients admitted in this period after application of inclusion/exclusion criteria. Namely, 8725 patients with a length of stay in the ICU of less than 10 days and 111 patients who received dialysis in the first five days of ICU admission were excluded from analysis.

Diuresis is measured in 2 hour intervals, while creatinine is measured one, two or exceptionally three times a day. These measurements are performed by hand, so there exists some variance in the intervals between succeeding measurements. Also the interval between creatinine measurements is larger than the one between diuresis measurements. However, the input time series needs to contain measurements over regular time intervals and these intervals must be the same for both input parameters. Therefore linear interpolation of the data is the very first preprocessing step.

The availability of both diuresis and creatinine measurements does not fully overlap. Measurements not within the overlapping interval are excluded from the data. Patients who do not have an overlapping interval of minimal 40 measurements are excluded from the study. After pre-processing, 830 patients are available with 60 interpolated measurements for both creatinine and diuresis. Figure D.2 visualizes these interpolated creatinine and diuresis measurements, expressed as milliliter/hour (ml/hr), for two patients. The patient in Figure 2a needed dialysis between five and ten days after ICU admission and the patient in Figure 2b did not. The interval between these measurements is 1 hour, so the data consists of a 60 hour period somewhere in the first 3 days of the patient's stay in the ICU. 62% of the patients were male and the mean age of the study population was 58.6

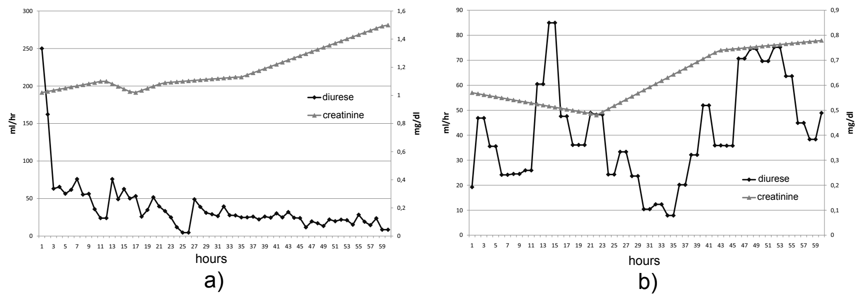


Figure D.2: ***Interpolated creatinine and diuresis measurements for an example patient who a) needed dialysis between five and ten days after ICU admission b) who did not need dialysis.***

years. The selected population had a total mortality rate of 17% and the mean Simplified Acute Physiology Score (SAPS) II score was 37.2. 82/830 (9.9%) patients needed dialysis between the fifth and tenth day after admission, while the remaining 748/830 (90.1%) patients did not need dialysis during that period.

D.3 Classifiers

In this section we discuss the feature extraction and selection methods and the classifiers under study. The time series are classified using support vector machines, the naive Bayes classifier and echo state networks. Prior to the use of SVM and the NB classifier, feature extraction and selection is required to reduce the number of input features and thus alleviate the effect of curse of dimensionality [33]. This way, we can compare the performance of two traditional classifiers - a sophisticated and a simple one - and the recent classifier based on ESN.

D.3.1 Feature extraction and selection

Classical classification techniques, such as the SVM and NB classifier, have been designed with a static data model in mind and are not suitable for coping with the dynamic nature of time series. The performance of the SVM and NB classifiers suffers from a large number of features if not all the features are of the same type and of equal importance [33]. This is the case in the medical problem addressed in this research as it consists of two types of features, namely diuresis and creatinine values. 60 interpolated measurements for both diuresis and creatinine are used as features. Not all these measurements are equally important as expert opinion reveals that the tails of the time series, i.e., later measurements, contain more information than the start of the series.

An inclusion of a large number of features in the SVM and NB classifiers leads to "the curse of dimensionality" [33, 34], which is associated with the following shortcomings:

- As the input dimensionality increases, the computational complexity and memory requirements of the model increase, which in turn increases the time to build the models.
- As the input variables increase, the number of training samples required also increase.
- Misconvergence and poor model accuracy may result from the inclusion of irrelevant inputs due to an increase in the number of local minima present in the error surface.
- Interpreting complex models is more difficult than interpreting simple models that give comparable results.

Feature extraction, which generates additional features from the time series, and feature selection, which selects the most appropriate features and thus reduces the amount of input features, helps to improve the performance of learning models by [35]:

- Alleviating the effect of the curse of dimensionality
- Enhancing generalization capability
- Speeding up the learning process and
- Improving model interpretability.

To make sure that all the information contained in the time series is captured, extensive feature extraction is applied for the SVM and the NB classifier. Features are therefore extracted that capture the overall properties of the time series and the correlation between the different measurements in the time series. For each time series the minimum, maximum, mean, median, 25^{th} percentile, 75^{th} percentile, standard deviation (stdev), the linear regression ($y = ax + b$) coefficients a and b and the area under the curve (AUC) are calculated. This results in 10 features per time serie.

As mentioned previously, expert opinion reveals that the tails of the time series contain more information than the start of the series. We therefore repeat the feature extraction multiple times for reduced time series. The 10 features are extracted for the full time series, the 59 last values of the time series, the 58 last values of the time series, ..., and the 2 last values of the time series. This results in $59 * 10 = 590$ extracted features per input parameter, or 1180 extracted features in total. Finally we add the measurements themselves to the extracted feature set as well, which results in $1180 + 2 * 60 = 1300$ features.

Feature selection needs to be performed on these 1300 features to select the most useful ones for the NB and SVM classifiers. Ideally, a brute-force search is performed in which the classification performance of each possible combination of features is tested and the best combination is selected. Brute-force feature selection is however very resource-intensive. As the number of possible feature combinations for 1300 features is nearly endless, namely ($2^{1300} - 1$) possible combinations, the required computation time would be virtually infinite.

To boost the performance, a greedy feature selection algorithm is used which iteratively adds the feature that improves prediction the best out of a set of features that show little multicollinearity with the already selected set of features. This approach is similar to the one used by Langley and Sage [36], but in each iteration we filter the set of candidates so that it contains only features that are not collinear with the already selected set. This drastically reduces the size of the set of candidate features in each iteration and therefore speeds up the feature selection process. Detection of multicollinearity is done using the common rule of thumb: $variance\,inflation\,factor > 5$ [37]. The classifier used in this hybrid filter-wrapper method [35] is the NB classifier.

All data is globally scaled to the $[-0.9, 0.9]$ interval. Scaling features to a fixed interval is necessary to avoid favoring a feature only because it has the largest scale. The bounds -0.9 and 0.9 are chosen instead of -1 and 1 to avoid excessive weight saturation in the recurrent artificial neural network.

D.3.2 Support vector machines

As first discussed by Cortes and Vapnik [38], a SVM tries to separate positive and negative examples in a multi-dimensional space by a hyperplane.

Assume that the training data is labeled as $\{\mathbf{x}_i, y_i\}, i = 1, \ldots, l, y_i \in \{-1, 1\}$, $\mathbf{x}_i \in \mathbf{R}^d$. The points $\mathbf{x}$ that lie on the hyperplane satisfy the equation $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w}$ is normal to the hyperplane. d_+ and d_- are the shortest distances from the separating hyperplane to the closest positive and negative example. The margin of the separating hyperplane is then defined as $d_+ + d_-$.

The SVM discussed by Cortes and Vapnik [38] was a linear classifier. For the linearly separable case, the SVM searches for the hyperplane that separates the data from the two classes with maximal margin [39]. This search can be formulated as an optimization problem, where

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \, \alpha_j \, y_i \, y_j \, \mathbf{x}_i \cdot \mathbf{x}_j \tag{D.1}$$

is maximized, subject to

$$\sum_i \alpha_i\, y_i = 0, \text{for } \alpha_i \geq 0 \tag{D.2}$$

with α_i being the Lagrangian multipliers for each training example. Given the α_i, the solution is given by

$$\mathbf{w} = \sum_i \alpha_i\, y_i\, \mathbf{x}_i \tag{D.3}$$

The examples for which $\alpha_i > 0$ are called support vectors. All other example have $\alpha_i = 0$.

When the positive and negative examples are not linearly separable, an additional condition needs to be added:

$$0 \leq \alpha_i \leq C \tag{D.4}$$

This gives the α_i een upper bound of C.

Switching to the non-linear case can be done by using the kernel-trick [40]. Notice that the data appears in the training problems, see Equation D.1, only in the form of dot products $\mathbf{x}_i \cdot \mathbf{x}_j$. If the data is mapped to some other Euclidian space *H*, using a mapping $\Phi : \mathbf{R}^d \mapsto H$, the training problem can be solved in *H* by replacing $\mathbf{x}_i \cdot \mathbf{x}_j$ by $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. If there is a kernel function *K* such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, then only *K* needs to be used in the training algorithm and it never needs to be explicitly known what Φ is. An example of such a kernel function and the one which was used in this study is the Radial Basis Function (RBF) kernel function. Rüping [41] showed that the RBF kernel performs very well on different types of time series and learning tasks. The RBF kernel function has the following definition:

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma\, \|\mathbf{x}_i - \mathbf{x}_j\|^2) \tag{D.5}$$

This results in a training algorithm with only two parameters, namely C and γ. For a more detailed introduction to SVMs, we refer to Burges [42]. SVMs have been successfully applied to perform time series prediction and prediction on real problems in different engineering fields [41, 43–45].

The libSVM [46] support vector machine implementation is used in this study. The C and γ parameters were optimized using parameter sweeps during each experiment, as is further detailed in Section D.4.

D.3.3 Naive Bayes classifier

The Naive Bayes Classifier is based on the application of Bayes' theorem, which relates the conditional and marginal probabilities of events A and B:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{D.6}$$

where $P(A)$ is the prior probability of A, $P(B)$ is the prior probability of B, $P(A|B)$ is the posterior probability of A and $P(B|A)$ is the posterior probability B.

A custom Java implementation of the Naive Bayes classifier is used in this study. This Naive Bayes classifier estimates the prior probability of class A as

$$P(A) \approx \frac{\text{\#items of class A in the training set}}{\text{total\#items in the training set}} \tag{D.7}$$

When a previously unseen example X is presented to the classifier, the likelihood of class A is estimated as

$$Li(A) \approx \frac{\text{\#items of class A in the training set in the neighborhood of X}}{\text{total\#items of class A in the training set}} \tag{D.8}$$

Assuming that each feature is conditionally independent of every other feature, the posterior probability that a previously unseen example X belongs to class A can be estimated as

$$P(X = A) \approx P(A) \times Li(A) \tag{D.9}$$

The number of examples in the training set that constitute the neighborhood of a previously unseen sample X, denoted by parameter k, is the only configurable parameter of the used Naive Bayes implementation. Parameter sweeps were performed to determine the optimal value for k per experiment, as is further detailed in Section D.4.

It can be noted that the Naive Bayes classifier is based on applying Bayes' theorem with strong independence assumptions. However, empirical results show that it performs surprisingly well in many domains containing clear feature dependencies [47]. Zhang [48] shows that the feature dependence distribution plays a crucial role in the explanation of this behavior and that sufficient and necessary conditions for the optimality of Naive Bayes can be formulated.

D.3.4 Echo state networks

The key idea in reservoir computing [14] is to feed time series to a reservoir, thereby modeling the dynamics of the system which generates the time series. The reservoir is then read by a readout function in order to make predictions using the constructed model. When training the model, only the readout function is modified, the complex dynamic modeling behavior of the reservoir is left unchanged.

In ESN [12], the reservoir consists of a recurrent artificial neural network with sigmoid activation functions and the *echo state property* which ensures good modeling abilities. A recurrent artificial neural network is said to have the echo state property when its state is uniquely determined by the input time series. This implies the *state forgetting property*: the initial state of the reservoir has no impact on the state after feeding a - possibly infinite - time series. Although it is not yet clearly understood how it exactly works, the reservoir acts as a short-term fading memory [17], which means in practical applications that the most recent input of the network has the largest impact on the prediction outcome. The readout function used in ESN is a linear classifier.

The general layout of an ESN is illustrated in Figure D.1. It consists of k input nodes, n reservoir nodes, and l output nodes. Each node is a perceptron with a sigmoid activation function. The state of each node at a given time is the weighted sum of the last fed inputs, namely

$$\mathbf{x}[t+1] = (1-\mu)\mathbf{x}[t] + \mu f(\mathbf{W}\mathbf{x}[t] + \mathbf{W}^{in}\mathbf{u}[t]) \qquad \text{(D.10)}$$

where $\mathbf{x}[t]$ denotes the network state at time t and $\mathbf{u}$ is the input matrix. Leaky integrator neurons are used to optimize the leak rate μ of the reservoir so that it can perfectly match the timescale of the input data. For every sample, $\mathbf{x}[0]$ is initalized as 0. The weights in the ESN are represented in weight matrices. The $k \times n$ matrix $\mathbf{W}^{in}$ contains the weights between the input and reservoir nodes and the $n \times n$ matrix $\mathbf{W}$ contains the recurrent weights between the reservoir nodes. The spectral radius λ_{max} is defined as the largest absolute eigenvalue of the matrix $\mathbf{W}$. It has been shown that reservoirs whose spectral radius is larger than one ($|\lambda_{\text{max}}| > 1$) do not have the echo state property, but in practice the spectral radius is chosen close to one to achieve a suitable dynamic response [12]. Zero weights are the equivalent of the absence of connections. Feedback connections from output nodes to reservoir nodes and connections from input nodes directly to output nodes are optional.

By using Equation D.10 the echo state network can be recursively simulated using the training data D_{train}. After each sample of the training data is simulated, the $|D_{train}|$ reservoir state matrices are concatenated in a large state matrix $\mathbf{A}$. Because an ESN is a dynamical system, it takes some time before the full effects of the input are visible in the reservoir states. Therefore, the initial states containing the transient effects are discarded which is known as warm-up drop. The number of states that is discarded is determined by the warm-up drop parameter α.

Different methods can then be used to train the linear readout function, and thus to determine the elements of the $(k+n+l) \times l$ output weight matrix $\mathbf{W}^{out}$, which contains the weights between the reservoir nodes and the output nodes. A complete overview and discussion of the different available techniques reported in literature for training the readout function of the reservoir can be found in Lukoševičius and Jaeger [23]. As the medical problem under study does not require on-line adaptation of the model, batch learning can be performed. In batch mode, the most recommended and used method is ridge or Tikhonov Regression [49], as it has the lowest computational cost, while still allowing to perform regularization. Ridge regression introduces a regularization parameter λ. In addition to improving the numerical stability, the regularization in effect reduces the magnitudes of entries in $\mathbf{W}^{out}$, thus mitigating sensitivity to noise and overfitting. However, because Fisher weighting is also used in this study to deal with the unbalanced data set, as further explained in the last two paragraphs of this section, ridge regression could not be used as this combination is not implemented in the Reservoir Computing Toolbox (RCToolbox) [50]. In this study, the RCToolbox is used to run the ESN experiments. However, using ridge regression is equivalent with using least squares regression [51] with noise. So, in this study, $\mathbf{W}^{out}$ is trained by performing least squares regression on the matrix $\mathbf{A}$, using the desired output matrix $\mathbf{y}$ as the right-hand side. Thus, the matrix $\mathbf{W}^{out}$ is computed that satisfies the equation:

$$\mathbf{W}^{out} = min_{\mathbf{W}} \left\| \mathbf{A} \times \mathbf{W} - \mathbf{y} \right\|^2 . \tag{D.11}$$

In practice, this equation can be computed in a single step by using the Moore-Penrose generalized matrix inverse, or pseudo-inverse, of the matrix $\mathbf{A}$ [52]. This provides least squares regression with a similar numerical stability as ridge regression. Gaussian noise is added to the matrix $\mathbf{A}$ in order to control the trade-off between model complexity and generalization capability (avoid overfitting). This guarantees that the model is complex enough to accurately model the underlying system, but not too complex such that it becomes sensitive to the noise in the samples. Similar to ridge regression, the amount of noise is determined by a regularization parameter λ.

Other methods that are sometimes used in literature to train the linear readout function are weighted regression and evolutionary search [53]. The first uses weights to emphasize some time steps t over others. As this study wanted to evaluate how well the ESN performed on the time series without using domain expert knowledge, this method was not used. State-of-art evolutionary methods are able to achieve the same level of precision for supervised tasks as with the best application of linear regression. However, their computational cost is much higher.

Finally, the output of the reservoir can be computed as follows:

$$\hat{\mathbf{y}}[k] = \mathbf{W}^{out}\mathbf{x}[k] \tag{D.12}$$

where $\hat{\mathbf{y}}$ is the actual output of the reservoir system.

As mentioned previously, the RCToolbox is used to run the ESN experiments. As the original time series, and thus not the extracted features, are used as input for the ESN, a reservoir with $k = 2$ input nodes and $l = 1$ output nodes is initialized. The elements of the input weight matrix $\mathbf{W}^{in}$ are drawn from the discrete set $\{-0.1, 0.1\}$ with equal probabilities. The density of the input weight matrix is 10%, which means that 10% of the weights are non-zero. The elements of the reservoir weight matrix $\mathbf{W}$ are drawn from a Gaussian distribution. The density D of the reservoir weight matrix is chosen as $d = 20\%$. The optimal value for the regularization parameter λ is determined by performing a brute-force grid search of the parameter space with cross-validation.

Output-to-output connections are not used. Input-to-output connections are used to enable a direct linear mapping of the input.

The RCToolbox allows performing parameter sweeps to find the optimal values for the various parameters of an ESN, namely the leak rate μ, the number of reservoir nodes n, the spectral radius $\lambda_{\max}$ and the warm-up drop parameter α. These optimal values are found by performing a sensitivity analysis for each parameter. This means that the values for this parameter are varied while all other parameter settings of the ESN are left unchanged. The parameter value which results in the best average performance of the ESN is chosen.

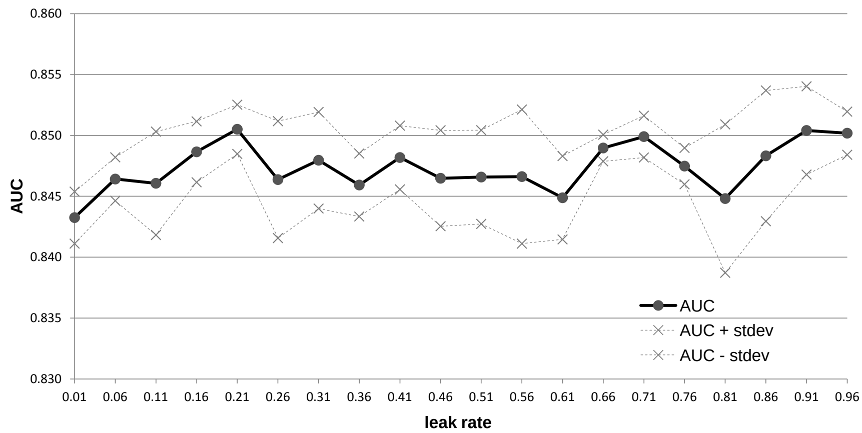


Figure D.3: ***Sensitivity analysis of the leak rate of the reservoir.*** *Dots and crosses are measured values. Lines are interpolated values. The area under the receiver operating characteristic (ROC) curve (AUC) is a performance measure. The solid line is the average performance in 30 runs. The dotted line denotes the observed standard deviation.*

The sensitivity analysis of the leak rate μ is visualized in Figure D.3. This figure shows the observed average performance and its standard deviation in 30 runs for leak rate values between $\mu = 0.01$ and $\mu = 1$. Higher performance values are better. For a more detailed explanation of the performance measure, see Section D.3.5. Different runs with the same settings result in different performance results because the data is randomly divided among the folds and the reservoir is randomly initialized. In theory, performance should not depend on these random circumstances. In practice, the dependence should be minimized. For example because of the limited amount of available data, there will always be a certain amount of dependence on how exactly the data is divided among the folds. In this problem setting, the best average performance and the smallest deviation in performance is aimed at. From Figure D.3 it is clear that adjusting the leak rate does not boost the performance of the ESN significantly. Likely this is due to the fact that in this case, the optimal parameters of the ESN are outside the usual range: for the high total input to the reservoir used here, the reservoir acts more like a static kernel rather than a dynamical system. As a consequence, the leak rate is chosen to be the value $\mu = 0.01$. This is the default value for the leak rate in the RCToolbox. This means that the reservoir will work very slowly, implementing a low-pass filter.

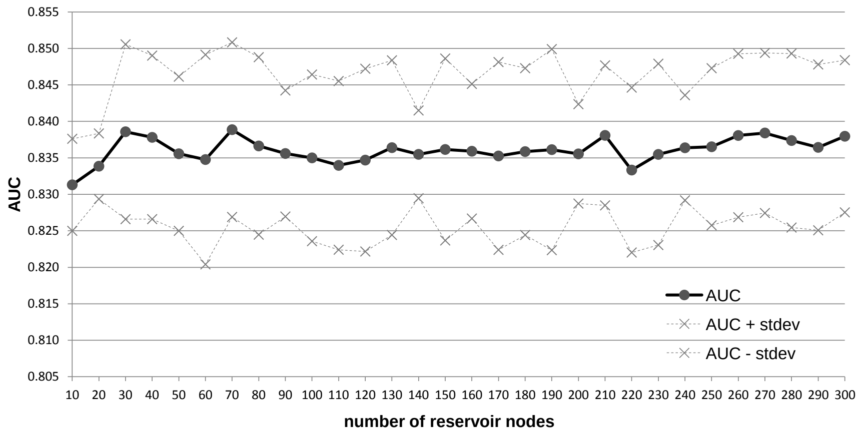


*Figure D.4: **Sensitivity analysis of the number of reservoir nodes.** Dots and crosses are measured values. Lines are interpolated values. The area under the receiver operating characteristic (ROC) curve (AUC) is a performance measure. The solid line is the average performance in 30 runs. The dotted line denotes the observed standard deviation.*

The sensitivity analyses of the number of reservoir nodes n and the spectral radius λ_{max} are shown in Figures D.4 and D.5. These figures show the observed average performance and its standard deviation in 30 runs for number of reservoir nodes between $n = 10$ and $n = 300$ and for spectral radius values between $\lambda_{\text{max}} = 0.1$ and $\lambda_{\text{max}} = 1.5$. From Figures D.4 and D.5 it can be derived that adjusting the number of reservoir nodes n or the spectral radius λ_{max} also had minimal effects on the performance of the ESN. As mentioned previously, a spectral radius close to one should be chosen to achieve a suitable dynamic response and to guarantee that the echo state property holds. Therefore, the spectral radius is chosen to be the value $\lambda_{\text{max}} = 0.99$. The weights are rescaled so that the spectral radius λ_{max} is set to this value. The number of reservoir nodes was chosen to be $n = 70$, as this was the parameter value with the highest average AUC across all the runs.

Finally, the warm-up drop parameter α is optimized by performing a sensitivity analysis. The observed average performance and its standard deviation in 30 runs for warm-up drop values between $\alpha = 0$ (no warm-up drop) and $\alpha = 59$ (only the last time point remains) are plotted in Figure D.6. From Figure D.6 it is clear that adjusting the warm-up drop parameter significantly boosts the performance of the ESN. A warm-up drop of $\alpha = 56$ first time steps of the time series leads to the best

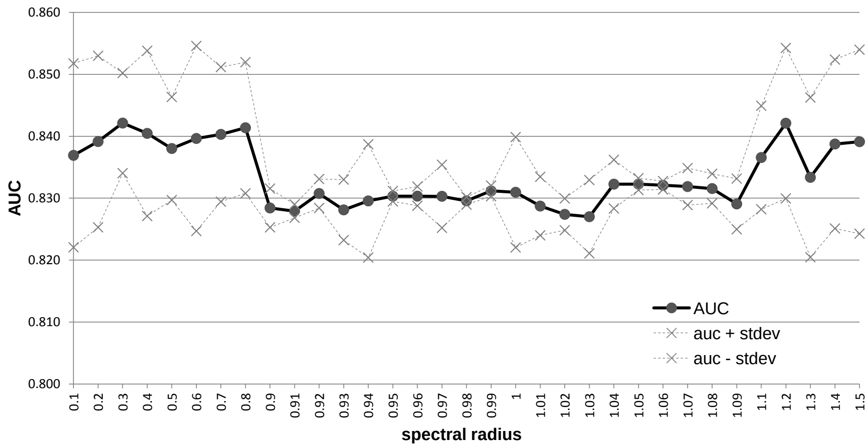


Figure D.5: ***Sensitivity analysis of the spectral radius.*** *Dots and crosses are measured values. Lines are interpolated values. The area under the receiver operating characteristic (ROC) curve (AUC) is a performance measure. The solid line is the average performance in 30 runs. The dotted line denotes the observed standard deviation.*

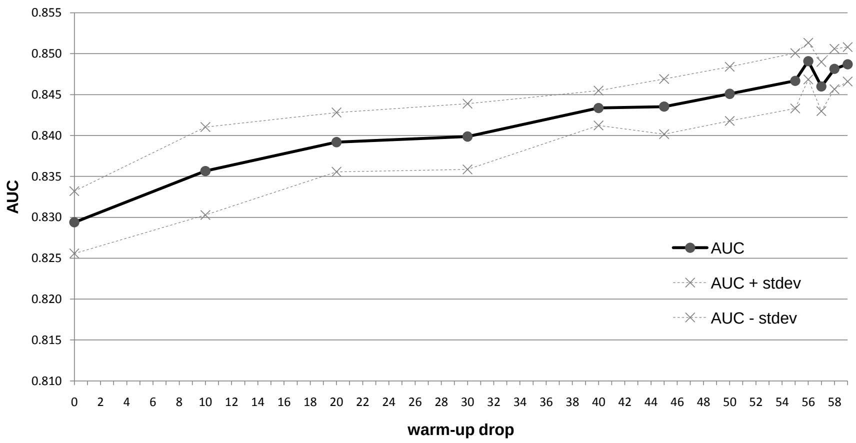


Figure D.6: ***Sensitivity analysis of the warm-up drop parameter of the echo state network.*** *Dots and crosses are measured values. Lines are interpolated values. The area under the receiver operating characteristic (ROC) curve (AUC) is a performance measure. The solid line is the average performance in 30 runs. The dotted line denotes the observed standard deviation.*

performance results. This corresponds with the opinion of the domain experts that the tail of the time series contains more information than the start of the series.

As can be noted from Section D.2, the dataset is unbalanced. There are a lot more examples of patients who did not receive dialysis between the fifth and tenth day after admission than there are patients that did (748 vs. 82 of the 830 patients). This unbalance will have an effect on the generalization capabilities of the classifiers. Since the read-out is trained using regression, the separating hyperplane will shift towards the class centers that are most present in the dataset (the threshold will not be zero). This is undesirable as one wants the hyperplane to lie in the middle between the two classes (threshold equal to zero). To achieve this, Fisher labeling is applied [54].

Assume, that the positive class has n_1 examples and the negative class has n_2 examples, then Fisher labeling relabels these classes from the usual $[-1, 1]$ for positive and negative examples respectively to $[(n_1 + n_2)/n_1, (n_1 + n_2)/n_2]$. In this way, the class labels reflect the unbalance of the number of examples in each class. This guarantees that the shifting of the hyperplane is undone. Thus for this dataset, the Fisher labeling relabels the classes to $[830/82, 830/748]$.

D.3.5 Performance evaluation

Each of the 3 used methods outputs a prediction score. The SVM and the NB output a prediction score per sample. The ESN, on the other hand, outputs a prediction score per time point in the time series. As the warm-up drop parameter α is set to 56, only 4 time points remain and thus 4 prediction scores are outputted by the ESN per sample. These are summarized to one prediction score per sample by taken the mean of these 4 values. In contrast with a categorical prediction - class A versus class B - a prediction score is a value $x \in \Re$ in the interval $]-\infty, +\infty[$. The sign of x corresponds to a class while the magnitude of x reflects the estimated probability of actually belonging to that class. By varying the prediction threshold, different classifiers can be constructed. These classifiers vary from one that classifies all patients into one class to one that classifies all patients into the other class.

The correctness of a classification can be evaluated by computing the number of true positives (TP, positive examples classified as positive), true negatives (TN, negative examples classified as negative), false positives (FP, negative examples classified as positive), and false negatives (FN, positive example classified as negative) respectively. The most often used measures for binary classification based on these values are Accuracy, Precision, Sensitivity (Recall), Specificity, F-Score and the area under the ROC curve (AUC) [55]. These measures differ in their ability to preserve their value under a change of the number TP, TN, FP, and/or FN. A measure is invariant if its value does not change when one or more of the TP, TN, FP, or FN values change. This inability can be beneficial or adverse, depending on the goal of the classification task. More information about the different performance measures for classification can be found in Sokolova and

Lapalme [55].

For the medical problem under scrutiny, we are interested in the overall performance of the classifier, i.e., interested in the performance of the classifier on both identifying and correctly classifying positive and negative examples. In other words, it is equally important to correctly identify whether a patient will receive dialysis or not between five and ten days after admission in the ICU. Precision, Recall and F-Score are invariant to changes in the number of TN. These measures thus do not acknowledge the ability of the classifiers to correctly identify negative examples. In contrast, Specificity is invariant to changes in the number of TP. This measure thus does not acknowledge the ability of the classifiers to correctly identify positive examples. Consequently, two measures remain that are non-invariant to changes to the number of TN and TP, namely AUC and accuracy. However, the accuracy is invariant to the distribution of classification results because it does not distinguish TP from TN and FN from FP. This measure is thus not trustworthy when using unbalanced data sets. The AUC is non-invariant to the distribution of classification results, which makes it a good measure for comparing classifiers on unbalanced data sets, such as the one used in this study.

The AUC is calculated based on the Specificity and Sensitivity performance measures of the classifier. Sensitivity measures the proportion of actual positive examples, i.e., patients needing dialysis, which are correctly identified by the classifier as follows:

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \tag{D.13}$$

In contrast, Specificity measures the proportion of actual negative examples, which are correctly identified by the classifier as follows:

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{D.14}$$

Plotting *Sensitivity* versus $(1-$*Specificity*$)$ for all these classifiers, results in the so called receiver operating characteristic (ROC) curve [56]. The area under this ROC curve (AUC) is an estimation of the probability that a positive patient receives a higher prediction score than a negative patient by the classification method under study. An AUC value of 1.0 indicates a classifier that perfectly separates positives from negatives, while a classifier that randomly classifies patients as positive or negative corresponds to $AUC = 0.5$. All other classifiers will result in $0.5 < AUC < 1.0$.

A two-sample t-test is used to determine whether an observed difference in AUC is random or real. A p-value expresses the probability of having a test statistic at least as extreme as the one that was actually observed, assuming that the null-hypothesis is true. The lower the p-value, the less likely the result, and consequently the more statistically significant the result is. A result is statistically significant if it is unlikely that it occurred by chance. Generally, the null hypothesis is rejected if the p-value is smaller than or equal to the significance level, α.

In this paper, the test statistics are the average AUCs across the 30 runs for each classifier. The null-hypothesis in these tests is that both average AUCs are equal. The significance level α is chosen to be 0.05, which expresses that results that are only 5% likely or less are deemed extraordinary, given that the null hypothesis is true.

Since we test 3 average AUCs for equality, the significance level α must be corrected for multiple testing. This can be done by applying Dunn-Šidák correction [57], that is

$$\alpha_{\text{cor}} = 1 - (1 - \alpha)^{1/C}, \tag{D.15}$$

where α is the chosen significance level, α_{cor} is the corrected α-value, and C is the number of tests. The null-hypothesis in this test is that both average AUCs are equal. Thus the corrected significance level, with whom the p-values are compared, is

$$\alpha_{\text{cor}} = 1 - (1 - 0.05)^{1/3} = 0.016952. \tag{D.16}$$

When choosing a prediction threshold, we can select the value where the balanced accuracy of the classifier is the highest. We define balanced accuracy as follows:

$$\text{balanced accuracy} = \frac{Sensitivity + Specificity}{2}. \tag{D.17}$$

Using maximum balanced accuracy prevents favoring a classifier that always outputs the majority class in the case of heavily unbalanced data sets such as the one used in this study. If the classifier performs equally well on either class, this term reduces to the conventional accuracy, i.e., the number of correct predictions divided by the total number of predictions. In contrast, if the conventional accuracy is above chance only because the classifier takes advantage of an imbalanced test set, then the maximum balanced accuracy will drop to chance.

As can be seen, the AUC gives us a global view on the quality of the constructed classifiers, while the maximum balanced accuracy is an indication of the best prediction accuracy we can expect. Moreover, the AUC is well-known and much used performance measure of binary classification tasks within the medical domain [58]. Both the AUC and maximum balanced accuracy are invariant to a uniform change of positive and negative examples in the data set. This means that these measures are stable with the respect to the uniform increase of the data size. As in our medical problem, the proportion of representatives for the positive and negative class will remain stable across different data sizes, these measures are a good choice. We will also look at the required execution time, which is a measure for the computational complexity of the methods under study.

D.4 Problem setting

To summarize, we compare the classification performance of 3 methods on the given problem. The performance measures are AUC and maximum balanced accuracy, which are determined for each of the methods using cross-validation. The computational complexity of the methods is compared through their required execution times. All these tests were performed on the same machine - Advanced Micro Devices (AMD) Athlon 64 X 2 Dual Core Processor, 3000 megahertz (MHz) Central Processing Unit (CPU), 2 Gigabyte (GB) of Random-Access Memory (RAM) - under exactly the same conditions.

The input data consists of 2 time series per patient. Each time series consists of 60 linear interpolated values, which are constructed out of the original patient data. For the ESN method, no further preprocessing of the data is necessary. Prior to the use of SVM and the NB classifier, feature extraction and selection and global rescaling of the data is required.

Several parts of the algorithms under study have a stochastic nature. Examples are the random division of the available data into folds and the random initialization of the reservoir weights in the ESN. To avoid faulty interpretation of results that origin from a coincidental odd configuration, the experiments are repeated 30 times, each time using another random initialization.

The pre-processing phase of the SVM and NB classifier, consisting of the feature extraction and selection process and global rescaling of the data, is also subject to random factors, for example, the random division of the available data into folds. Moreover, there are several multicollinear features. In each iteration of the feature selection, the set of candidates is filtered so that it contains only features that are not collinear with the already selected set. Which of the multicollinear features thus ends up in the selected set is also subject to the random initialization of the feature selection. Therefore, this pre-processing phase is also repeated for each run of the SVM and NB classifier.

Consequently, the data set that is used as input for the NB and SVM classifiers is different in every run. To determine the optimal values for the parameter k of the NB classifier and parameters C and γ of the SVM classifier parameter sweeps thus need to be performed for each of the 30 runs. For each parameter, the value is selected that achieves the highest performance for the classifier in that run. Consequently, different parameter values are obtained for the NB and SVM classifiers in each run. The optimal value of the parameter k of the NB classifier across the 30 runs ranges from $k = 29$ to $k = 47$ and is on average $k = 40$. The optimal value of the parameters C and γ of the SVM classifier across the 30 runs range from $C = -4.12$ to $C = 23.65$ and $\gamma = -22.05$ to $\gamma = -10.57$ and are on average $C = 17.66$ and $\gamma = -17.58$.

D.5 Results

	best	average	stdev	CI 95%	CI 99%
ESN - optimized	*0.854*	*0.849*	*0.002*	*0.001*	*0.001*
ESN - default	*0.804*	*0.799*	*0.003*	*0.001*	*0.001*
SVM + FE + FS	*0.857*	*0.838*	*0.021*	*0.007*	*0.010*
NB + FE + FS	***0.885***	***0.874***	*0.006*	*0.002*	*0.003*

Table D.1: Observed area under the curve (AUC) in 30 *runs using* 3 *different classification methods: the echo state network (ESN), the support vector machine (SVM) and the naive Bayes classifier (NB). The latter two are preceded by a pre-processing phase, consisting of the feature extraction (FE) and feature selection (FS) process and global rescaling of the data.*

	best	average	stdev	CI 95%	CI 99%
ESN - optimized	*0.803*	*0.795*	*0.002*	*0.001*	*0.001*
ESN - default	*0.746*	*0.742*	*0.003*	*0.001*	*0.001*
SVM + FE + FS	*0.812*	*0.784*	*0.019*	*0.007*	*0.009*
NB + FE + FS	***0.826***	***0.809***	*0.009*	*0.003*	*0.004*

Table D.2: Observed maximum balanced accuracy in 30 *runs using* 3 *different classification methods: the echo state network (ESN), the support vector machine (SVM), and the naive Bayes classifier (NB). The latter two are preceded by a pre-processing phase, consisting of the feature extraction (FE) and feature selection (FS) process and global rescaling of the data.*

	SVM + FE + FS	NB + FE + FS
ESN - optimized	*0.0097*	$<$ *0.001*
SVM + FE + FS		$<$ *0.001*

Table D.3: P-values resulting from the tests for equality between the AUCs.

Table D.1 and Table D.2 show respectively the observed AUC and maximum balanced accuracy performance measures. The best maximum balanced accuracy and best AUC achieved across the 30 runs for each classification method are shown as well as the average value and its accompanying standard deviation (stdev) and Confidence Intervals (CI) at 95% and 99%. The performance measures for the ESN classifier are shown for both the configuration for which all the parameter values of the ESN were optimized through parameter sweeps and the default configuration which uses the default settings of the RCToolbox for the ESN. The default settings are a reservoir size $n = 100$, a leak rate $\mu = 0.01$, a scale factor $\lambda_{\text{max}} = 0.9$ and a warm-up drop $\alpha = 0$. Table D.3 contains the p-values that are obtained while testing the average AUCs for equality.

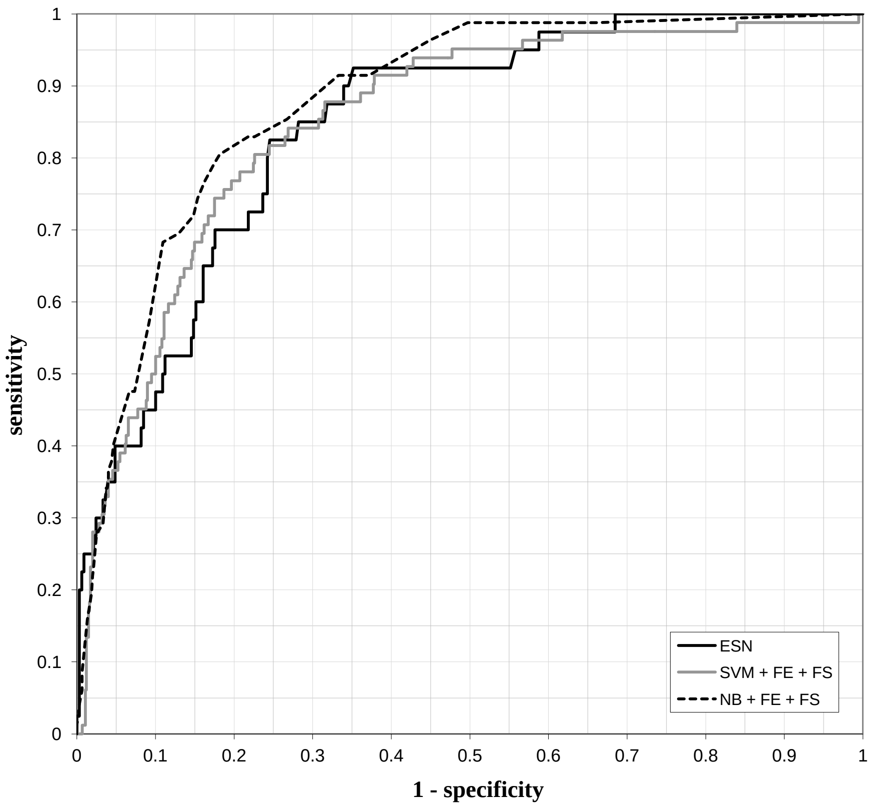


Figure D.7: ***The obtained ROC curves in run*** 2 ***for the echo state network (ESN), the support vector machine (SVM), and the naive Bayes classifier (NB).*** *The latter two are preceded by a pre-processing phase, consisting of the feature extraction (FE) and feature selection (FS) process and global rescaling of the data.*

Figure D.7 shows the obtained ROC curves in run 1. The obtained ROC curves in the other runs are very similar.

	average	stdev	CI 95%	CI 99%
ESN	***253.87ms***	*6.86ms*	*2.45ms*	*3.22ms*
SVM & NB	*3h 59m 55s 245.93ms*	*47m 6s 299.70ms*	*16m 51s 359.77ms*	*22m 9s 152.05ms*

Table D.4: Required computation time for the data pre-processing phase for the support vector machine (SVM), the naive Bayes classifier (NB) and the echo state network (ESN). SVM and NB share the same pre-processing phase, consisting of the feature extraction (FE) and feature selection (FS) process and global rescaling of the data.

As Table D.4 shows, the pre-processing phase preceding the support vector machines and naive Bayes classifier approach, which includes the loading and interpolating the data and performing feature extraction and selection, requires on average 3 hours (h) 59 minutes (m) 55 seconds (s) and 245.93 milliseconds (ms) of computation time. The pre-processing phase for the recurrent reservoir, which only includes loading and interpolating the data as no feature extraction and selection is needed, requires on average only 253.87 ms of computation time.

	average	stdev	CI 95%	CI 99%
ESN - default	***6m 35s 209.35ms***	*1s 247.55ms*	*446.42ms*	*586.70ms*
SVM	*14m 19s 936.23ms*	*5m 37s 934.51ms*	*2m 0s 926.09ms*	*2m 38s 923.82ms*
NB	*35m 37s 258.40ms*	*39s 154.80ms*	*14s 11.11ms*	*18s 413.72ms*
ESN - 1 parameter sweep	*6m 35s 117.86ms*	*2s 647.05ms*	*947.22ms*	*1s 244.86ms*

Table D.5: Required train time for the support vector machine (SVM), the naive Bayes classifier (NB) and the echo state network (ESN)

Table D.5 shows the computation time needed to train the three classifiers. The reported train time includes finding the optimal value for the size of the neighborhood k, see Equation D.8, for the NB classifier, for the C and γ parameters, see Equations D.4 and D.5, of the SVM classifier and the regularization parameter λ of the ESN classifier with default configuration. To reach the performance results of the Optimized ESN classifier, parameter sweeps need to be performed. The train time for performing one parameter sweep of the reservoir size, leak rate, scale factor or warm-up drop parameters of the ESN are also reported. Performing one sweep means that this parameter is set to 1 value (e.g. reservoir size = 300) and the ESN is trained. In practice, mainly the warm-up drop parameter needed to be sweeped to obtain the improved performance results of the Optimized ESN classifier.

	average	stdev	CI 95%	CI 99%
ESN	***0.030ms***	*0.009ms*	*0.003ms*	*0.004ms*
SVM	*0.033ms*	*0.183ms*	*0.065ms*	*0.086ms*
NB	*0.300ms*	*0.466ms*	*0.167ms*	*0.219ms*

Table D.6: Required test time for the support vector machine (SVM), the naive Bayes classifier (NB) and the echo state network (ESN).

Finally, Table D.6 visualizes the computation time needed to test the three classifiers with data about one patient.

D.6 Discussion

The p-values comparing the naive Bayes (NB) classifier combined with feature extraction (FE) & selection (FS), the support vector machine (SVM) combined with FE & FS and the echo state network (ESN) classifier are smaller than the

Dunn-Šidák corrected significance level $\alpha = 0.016952$, see Equation D.16. We therefore conclude that there is a significant difference between the average AUCs of the used methods observed at the 5% level. This means that the SVM + FE + FS, with an average AUC of 0.838, is the worst classifier. The NB + FE + FS has an average AUC of 0.874 and is thus the best classifier. The ESN classifier lies somewhere in the middle with an average AUC of 0.849. Inspection of Figure D.7, which shows the ROC curves, and Table D.2, which shows the observed maximum balanced accuracy, see Equation D.17, confirms this conclusion. However, the results of the NB classifier combined with FE & FS are biased as the feature selection method is a hybrid filter-wrapper method which also uses a NB classifier as classifier. Consequently, features selected by this hybrid filter-wrapper method are optimal for and best recognized by the NB classifier used in this feature selection method. If we then again apply a NB classifier on the selected features, the achieved results are slightly biased towards the NB classifier, since the selected feature set favors this type of classifier.

Based on the observed values of the performance measures we cannot definitely favor the ESN classifier. The picture changes when we look at the procedure followed for each method. The SVM and the NB classifier are designed for datasets where the data resides in an n-dimensional space as such. The longitudinal correlation along the different dimensions/parameters is not taken into account in any way. Therefore SVM and NB perform rather poorly when time series data is used unprocessed. To get satisfying results, we first must extract useful features based on the time series. This can be done in an automated way or by using domain knowledge of the problem at hand. Extracting features in an automated way often results in missing important characteristics of the data, while acquiring domain knowledge is a time consuming and often cumbersome activity. In this study we used a combined approach, exploiting the time saving properties of automated feature extraction and limiting the domain knowledge gathering to acquiring general properties of the data. The latter allows to steer the automated procedure, which avoids exploring useless regions in the search space. This approach still results in an enormous amount of candidate features, which makes a feature selection phase necessary as well. Furthermore, both feature extraction and feature selection phases combined require a considerable amount of computation time, namely on average approximately 4 hours (see Table D.4).

In the ESN approach, no feature extraction and selection is needed. The reservoir stores features from the input data and actually adds features to it, as we go from an input space from $k = 2$ dimensions to a reservoir space of $n = 70$ dimensions. Thus, by putting a reservoir between the input data and the readout, a lot more features are available to build the estimation on. The ESN consequently succeeds nicely in modeling the information contained in the time series data. It therefore needs on average less than a second of pre-processing time (see Table D.4) and no domain knowledge. Additionally, the reservoir algorithms are easy to implement, and existing rules of thumb suffice for acquiring a good performing configuration of the reservoir, as can be noted from the performance of the ESN with default configuration in Table D.1. Moreover, a simple linear regression classification suffices for determining the final classification results, where complex non-linear methods are required in the traditional approach.

Note that expert opinion states that in the data used the required information is mostly contained in the tail of the time series. This was explicitly taken into account during the pre-processing phase of the SVM and NB classifiers by extracting features from an increasingly shorter time series. Namely, the 10 features were extracted for the full time series, the 59 last values of the time series, the 58 last values of the time series, ..., and the 2 last values of the time series. If we study the features, which were selected during the feature selection phase, we see that mainly features of the shorter time series and linear regression coefficients were selected. However, for the ESN classifier this domain knowledge does not need to be taken explicitly into account. The ESN classifier takes it implicitly into account because of the fading short-term memory [17] characteristic of the ESN. This means that the most recent input of the network has the largest impact on the prediction outcome, which matches the domain knowledge that the most important information is contained in the tail of the time series. This explains the successful results.

The computation time for training the ESN classifier is also better than the other classifiers, as shown in Table D.6. However, additional time is needed to optimize the values of the various parameters of the ESN classifier through parameter sweeps. Optimizing the value of the warm-up drop parameter resulted in significant performance improvements. In practice, about 5 sweeps would have to be performed to obtain the optimal value for the warm-up drop parameter. Therefore, the train time for the different classifiers is comparable.

As can be derived from Table D.6, the test time of the SVM and ESN is also comparable. The computation time for testing the NB classifier is slightly higher on average, because the NB classifier takes into account each training sample when calculating the neighborhood of the testing sample. Since the data set used in this study is relatively small, the difference in test time between the NB classifier and the SVM and ESN classifiers is still negligible.

Since the ESN allows complex non-linear modeling in a simpler and computationally much more efficient way compared to the traditional approach while yielding a comparable classification performance, the authors believe that the ESN will play an important role in future analysis of medical time series data.

D.7 Conclusion

Medical data often consists of time series. This kind of data should be analyzed by specialized methods. The echo state network (ESN) is a recent method that was optimized to handle time series data. ESNs are easy to implement and to use, and do not require that feature extraction and selection is performed on the time series data before using it as input. We show the usefulness of ESN by using it to predict the need for dialysis between the fifth and tenth day after admission in ICU patients, and comparing the results to those acquired by using support vector machines (SVM) and the naive Bayes (NB) classifier combined with feature extraction (FE) and selection (FS). A hybrid filter-wrapper feature selection method is used with an NB classifier as classifier. Performance is measured by the area under the ROC curve and the maximum balanced accuracy.

Limitations of this study are that no extensive comparative study was performed between different feature selection methods that could be combined with the SVM and NB classifiers and the lack of comparison of the ESN to other classification methods which can directly process time series. Future work will further investigate these limitations by studying if the choice of the feature selection method significantly improves the performance of the SVM and NB classifiers on this medical classification task. Moreover, the performance of the ESN will be compared to other reservoir computing methods, such as liquid state machines and backpropagation decorrelation.

The results of this study showed statistically significant difference at the 5% level between the performance of ESN and the other two methods. The SVM + FE + FS had the worst performance, the NB classifier + FE + FS the best and the performance of the ESN lies in the middle. However, the results of the NB classifier + FE + FS are biased as the feature selection method is a hybrid filter-wrapper method which also uses a NB classifier. Moreover, its simplicity in usage, its ability to model and extract features without the need of domain knowledge, and its limited usage of computing time, make ESN the most suitable method for predicting the need for dialysis when using measured time series as input.

Future work will focus on applying the reservoir computing methods on a medical classification task which is not trivial for the medical experts, namely detecting whether a patient who has been admitted to the ICU has sepsis. Sepsis is the number one cause of death in the ICU.

Acknowledgment

Femke Ongenae would like to thank the IWT, Institute for the promotion of Innovation through Science and Technology in Flanders, for supporting this work through her PhD grant.

References

[1] E. Levin. *Hidden control neural architecture modeling of nonlinear time varying systems and its applications*. IEEE Transactions on Neural Networks, 4(1):109–116, 1993.

[2] K. Iso and T. Watanabe. *Speaker-independent speech recognition using a neural prediction model*. Electronics and Communications n Japan (Part III: Fundamental Electronic Science), 74(8):22–30, 1991.

[3] J. Tebelskis, A. Waibel, B. Petek, and O. Schmidbauer. *Continuous speech recognition by linked predictive neural networks*. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, Proceedings of the conference on Advances in Neural Information Processing Systems (NIPS), pages 199–205, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[4] J.-X. Xie, C.-T. Cheng, K.-W. Chau, and Y.-Z. Pei. *A hybrid adaptive time-delay neural network model for multi-step-ahead prediction of sunspot activity*. International Journal of Environment and Pollution, 28(3-4):364–381, 2006.

[5] A. Robinson. *An application of recurrent nets to phone probability estimation*. IEEE Transactions on Neural Networks, 5(2):298–305, 1994.

[6] L. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2):257–286, 1989.

[7] A. Graves and J. Schmidhuber. *Framewise phoneme classification with bidirectional LSTM and other neural network architectures*. Neural Networks, 18(5-6):602–610, 2005.

[8] E. Trentin and M. Gori. *Robust combination of neural networks and hidden Markov models for speech recognition*. Neural Networks, 14(6):1519–1531, 2003.

[9] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, New Jersey, 1994.

[10] H. Jaeger. *A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF, and the "echo state network"*. Technical Report GMD 159, German National Research Institute for Computer Science, 2002.

[11] W. Maass, T. Natschläger, and H. Markram. *Real-time computing without stable states: A new framework for neural computation based on perturbations.* Neural Computation, 14(11):2531–2560, 2002.

[12] H. Jaeger. *The "echo state" approach to analysing and training recurrent neural networks*. Technical Report GMD 148, German National Research Institute for Computer Science, 2001.

[13] J. Steil. *Online stability of backpropagation-decorrelation recurrent learning*. Neurocomputing, 69(7-9):642–650, 2006.

[14] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. *An experimental unification of reservoir computing methods*. Neural Networks, 20(3):414–423, 2007.

[15] H. Jaeger, W. Maass, and J. Principe. *Special issue on echo state networks and liquid state machines: Editorial.* Neural Networks, 20(3):287–289, 2007.

[16] R. Fisher. *Statistical methods for research workers*. Oliver and Boyd, Edinburgh, 1925.

[17] H. Jaeger. *Short term memory in echo state networks*. Technical Report GMD 152, German National Research Institute for Computer Science, 2002.

[18] H. Jaeger. *Adaptive nonlinear system identification with echo state networks*. In S. Becker, S. Thrun, and K. Obermayer, editors, Advances in Neural Information Processing Systems (NIPS), pages 593–600, Cambridge, MA, 2003. MIT Press.

[19] H. Jaeger and H. Haas. *Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication*. Science, 304(5667):78–80, 2004.

[20] W. Maass, T. Natschläger, and H. Markram. *A model for real-time computation in generic neural microcircuits*. In S. Becker, S. Thrun, and K. Obermayer, editors, Advances in Neural Information Processing Systems (NIPS), pages 213–220, Cambridge, MA, 2003. MIT Press.

[21] W. Maass, T. Natschläger, and H. Markram. *Computational Neuroscience: A Comprehensive Approach*, chapter Computational models for generic cortical microcircuits, pages 575–605. CRC-Press, Boca Raton, Florida, USA, 2004.

[22] T. Natschläger, H. Markram, and W. Maass. *A Practical Guide to Neuroscience Databases and Associated Tools*, chapter 9: Computer models and analysis tools for neural microcircuits, pages 123–128. Kluwer Academic Publishers, Boston, 2002.

[23] M. Lukoševičius and H. Jaeger. *Reservoir computing approaches to recurrent neural network training*. Computer Science Review, 3(3):127–149, 2009.

[24] W. Maass and C. Bishop. *Pulsed neural networks*. Bradford Books/MIT Press, Cambridge, MA, USA, 2001.

[25] W. Gerstner and W. M. Kistler. *Spiking neuron models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge, UK, 2002.

[26] U. D. Schiller and J. J. Steil. *Analyzing the weight dynamics of recurrent learning algorithms*. Neurocomputing, 63:5–23, 2005.

[27] B. Schrauwen, D. Verstraeten, and J. V. Campenhout. *An overview of reservoir computing: theory, applications and implementations*. In 15th European Symposium on Artificial Neural Networks (ESANN 2007), pages 471–482, 2007.

[28] T. Verplancke, S. Van Looy, K. Steurbaut, T. Benoit, F. De Turck, G. De Moor, and et al. *A novel time series analysis approach for prediction of dialysis in critically ill patients using echo-state networks*. BMC Medical Informatics and Decision Making, 4, 2010.

[29] B. Noris, M. Nobile, L. Piccini, M. Berti, E. Mani, M. Molteni, and et al. *Gait analysis of autistic children with Echo State Networks*. Parkinsonism & Related Disorders, 14(Suppl. 1):S70, 2008.

[30] P. Buteneers, B. Schrauwen, D. Verstraeten, and D. Stroobandt. *Real-time Epileptic Seizure Detection on Intra-cranial Rat Data using Reservoir Computing*. In M. Köppen, N. Kasabov, and G. Coghill, editors, 15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly (ICONIP 2008), pages 56–63, Berlin, 2008. Springer-Verlag.

[31] P. Buteneers, D. Verstraeten, P. van Mierlo, T. Wyckhuys, D. Stroobandt, R. Raedt, H. Hallez, and B. Schrauwen. *Automatic detection of epileptic seizures on the intra-cranial electroencephalogram of rats using reservoir computing*. Artificial Intelligence in Medicine, 53(3):215–223, 2011.

[32] S. Zeger, R. Irizarry, and R. Peng. *On time series analysis of public health and biomedical data*. Annual Review of Public Health, 27:57–79, 2006.

[33] M. H. Bowden G.J., Dandy G.C. *Input determination for neural network models in water resources applications. Part 1 - Background and methodology*. Journal of Hydrology, 301:75–92, 2005.

[34] N. Muttil and K.-W. Chau. *Machine learning paradigms for selecting ecologically significant input variables*. Engineering Applications of Artificial Intelligence, 20(6):735–744, 2007.

[35] I. Guyon and A. Elisseeff. *An Introduction to Variable and Feature Selection*. Journal of Machine Learning Research, 3:1157–1182, 2003.

[36] P. Langley and S. Sage. *Induction of selective Bayesian classifiers*. In R. de Mántaras and D. Poole, editors, Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94), pages 399–406, San Mateo, 1994. Morgan Kaufmann.

[37] M. Kutner, J. Neter, C. Nachtsheim, and W. Li. *Applied Linear Regression Models*. McGraw-Hill, New York, 2004.

[38] C. Cortes and V. Vapnik. *Support-vector networks*. Machine Learning, 20(3):273–297, 1995.

[39] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, Berlin, 1995.

[40] M. Aizerman, E. Braverman, and L. Rozonoer. *Theoretical foundations of the potential function method in pattern recognition learning*. Automation and Remote Control, 25:821–837, 1964.

[41] S. Rüping. *SVM kernels for time series analysis*. In R. Klinkenberg, S. Rüping, A. Fick, N. Henze, C. Herzog, R. Molitor, and O. Schrüder, editors, Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität (LLWA-01), pages 43–50, Dortmund, Germany, 2001. University of Dortmund.

[42] C. Burges. *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery journal, 2:121 – 167, 1998.

[43] J.-Y. Lin, C.-T. Cheng, and K.-W. Chau. *Using support vector machines for long-term discharge prediction*. Hydrological Sciences Journal, 51(4):599–612, 2006.

[44] A. Kampouraki, G. Manis, and C. Nikou. *Heartbeat time series classification with support vector machines*. IEEE Transactions on Information Technology in Biomedicine, 13(4):512–518, 2009.

[45] D. Zhang, W. Zuo, D. Zhang, and H. Zhang. *Time Series Classification Using Support Vector Machine with Gaussian Elastic Metric Kernel*. In 20th International Conference on Pattern Recognition (ICPR), pages 29–32, Piscataway, NJ, USA, 2010. IEEE Computer Society.

[46] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2012. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (Published: 2001, Accessed: 25 July 2012).

[47] P. Domingos and M. Pazzani. *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*. Machine Learning, 29:103–130, 1997.

[48] H. Zhang. *The Optimality of Naive Bayes*. In V. Barr and M. Zdravko, editors, Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, pages 562–567, Menlo Park, CA, 2004. AAAI Press.

[49] F. Wyffels, B. Schrauwen, and S. Dirk. *Stable Output Feedback in Reservoir Computing Using Ridge Regression*. In V. Kurkova-Pohlova and J. Koutnik, editors, 18th International Conference on Artificial Neural Networks (ICANN 2008), pages 807–818, Berlin, Heidelberg, 2008. Springer-Verlag.

[50] D. Verstraeten and M. Wardermann. *The Reservoir Computing Toolbox v2.0*, 2012. Software available at http://snn.elis.ugent.be/rctoolbox (Published: 2009, Accessed: 25 July 2012).

[51] A. Björck. *Numerical Method for Least Squares Problems*. SIAM, Philadelphia, PA, USA, 1996.

[52] R. Penrose. *A generalized inverse for matrices*. Mathematical proceedings of the Cambridge Philosophical Society, 51:406–413, 1955.

[53] F. Jiang, H. Berry, and M. Schoenauer. *Supervised and evolutionary learning of echo state networks*. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, 10th International Conference on Parallel Problem Solving from Nature (PPSN), pages 215–224, Berlin, Heidelberg, 2008. Springer-Verlag.

[54] R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd ed.)*, chapter Relation to Fisher's Linear Discriminant, page 654. Wiley Interscience, New York, NY, USA, 2001.

[55] M. Sokolova and G. Lapalme. *A systematic analysis of performance measures for classification tasks*. Information Processing and Management, 45(4):427–437, 2009.

[56] M. Zweig and G. Campbell. *Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine*. Clinical chemistry, 39(4):561–577, 1993.

[57] H. Abdi. *Encyclopedia of Measurement and Statistics*, chapter The Bonferroni and Šidák corrections for multiple comparisons, pages 1–9. Sage, Thousand Oaks, CA, 2007.

[58] M. Sokolova, N. Japkowicz, and S. Szpakowicz. *Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation*.

In A. Sattar and B.-h. Kang, editors, 19th Australian joint conference on Artificial Intelligence: advances in Artificial Intelligence (AI'06), pages 1015–1021, Berlin, Heidelberg, 2006. Springer-Verlag.