

Alma Mater Studiorum – Università di Bologna

**DOTTORATO DI RICERCA IN
BIOINGEGNERIA**

Ciclo XXIII

**Sistemi per il monitoraggio concorrente di parametri biometrici
e ambientali finalizzato alla valutazione di situazioni critiche**

Presentata da: Fabio Vergari

Coordinatore Dottorato

Prof. Angelo Cappello

Relatore

Prof. Claudio Lamberti

Esame finale anno 2011

Sommario

1	Introduzione	7
2	La telemedicina: motivazioni e stato dell'arte	11
2.1	Cenni storici.....	11
2.2	L'interoperabilità	13
2.2.1	Contesto	13
2.2.2	Sistema interoperabile.....	15
2.2.3	Ontologia.....	16
2.2.4	Smart Space	17
2.3	Struttura della tesi e contributi.....	17
3	Progetto e sviluppo di un sistema indossabile per la contestualizzazione del segnale ECG.....	19
3.1	Architettura del sistema.....	20
3.2	Una piattaforma multi sensore: l' iMote2	21
3.2.1	Sistema operativo.....	23
3.3	Acquisizione del segnale ECG	24
3.3.1	Front-end analogico	25
3.3.2	Sezione digitale	26
3.4	Considerazioni	27
4	Le tecnologie per l'interoperabilità delle informazioni	31
4.1	Ontologia	31
4.2	RDF e RDFS.....	32
4.3	OWL e RDF++	35
4.4	Ontologie esistenti.....	37

5	Spazi Intelligenti e piattaforme di supporto	39
5.1	Smart Space e Smart-M3	40
6	Disegno dell' ontologia.....	45
6.1	Metodologie per lo sviluppo di una ontologia.	45
6.2	Progetto dell' ontologia per Smart Space	48
6.3	Conclusioni.....	54
7	Progetto del prototipo di Smart Space per il	
	monitoraggio concorrente di parametri fisiologici ed ambientali	55
7.1	Progettazione di uno Smart Space.....	55
7.2	Prototipo di Smart Space.....	57
7.2.1	Motivazioni del prototipo.....	57
7.2.2	Requisiti del prototipo.....	58
7.2.3	Specifiche del prototipo.....	61
7.2.4	Architettura del prototipo	63
7.3	Conclusioni.....	69
8	Conclusioni	71
8.1	Raccomandazioni per lo sviluppo di un sistema interoperabile.	71
8.2	Sviluppi futuri.....	75
	Bibliografia.....	83

1 Introduzione

La ricerca e lo sviluppo di sistemi di monitoraggio, sistemi indossabili intelligenti e applicazioni in grado di abilitare servizi personalizzabili, in particolare in ambito sanitario, si sono sviluppati sempre più negli ultimi anni a livello internazionale. La principale motivazione delle applicazioni in ambito sanitario è la necessità di rispondere positivamente alla sfida finalizzata a ridurre i costi della spesa sanitaria mantenendo al tempo stesso un elevato livello della qualità dei servizi; questi ultimi devono essere resi facilmente accessibili ovunque con costi di gestione base sempre più bassi e con qualità del servizio sempre migliore, garantendo comunque sia l'accesso a tutti, sia il libero mercato, e quindi senza mai privilegiare un costruttore piuttosto che un altro; in questa ottica assumono sempre maggiore importanza i programmi di prevenzione cura e benessere personalizzati e assume sempre maggiore importanza il ruolo della tecnologia nella definizione delle politiche di settore [1].

Attualmente il sistema sanitario è sotto pressione dovendo rispondere a sfide che nascono da problematiche demografiche e socio-economiche. La popolazione sta invecchiando; la percentuale di persone con un'età superiore ai 65 anni crescerà dall'attuale 17 % al 25 % nel 2030 e la lunghezza di vita media è stimata crescere, raggiungendo nel 2030 81,4 anni rispetto i 75 odierni [2]. Nel tempo stesso si ha anche un aumento del numero di persone con malattie croniche e con qualche menomazione (vista, udito, problemi di destrezza, declino cognitivo). Proprio le malattie croniche rappresentano la più grande causa di morte precoce e disabilità; in particolare le malattie cardiovascolari sono la principale causa di morte in Europa e nei paesi industrializzati. I costi del servizio sanitario stanno aumentando da una media mondiale del 9 % del PIL nel 2005 all' 11 % atteso nel 2015. La maggior parte degli stati europei spende tra il 30 e il 40 % di tutti i costi sanitari per la popolazione anziana e cure a lungo termine. A tutte queste problematiche si aggiunge una barriera presente tra strutture mediche e medici specialistici che impedisce un efficiente scambio di informazioni.

Per risolvere queste problematiche, in molti stati il sistema sanitario è soggetto a riforme [3]; la vera sfida è fornire e controllare i servizi sanitari aumentandone l'efficienza e l'efficacia. Gli obiettivi sono controllare i costi riducendo i test e gli esami non necessari fornendo accesso a tutte le informazioni rilevanti del paziente promuovendo una integrazione nella raccolta dei dati su diagnosi e trattamenti. L'approccio comune per aumentare la qualità dei servizi sanitari consiste nel tentativo di fornire il servizio sempre e ovunque e nello spostare la mentalità da "come curare il paziente" a "come tenere le persone sane prevenendo malattie". Quindi, come già accennato, il focus si è spostato verso la prevenzione piuttosto che sulla cura, verso dimissioni più veloci e ricoveri più economici favorendo la riabilitazione a casa del paziente.

L'*Information and Communication Technology* (ICT) gioca un ruolo fondamentale in questo cambiamento. Nello specifico contribuisce a ridurre le distanze tra medico e paziente con servizi di telemonitoraggio; quest'ultimo è basato sull'integrazione di assistenza sanitaria, telecomunicazioni e tecnologie dell'informazione. Il telemonitoraggio abilita l'osservazione continua ed in tempo reale dello stato di salute di un paziente e offre l'opportunità di raccogliere e analizzare un elevato numero di dati riguardanti il paziente e la sua storia clinica. Inoltre l'ICT fornisce allo staff medico la possibilità di aumentare la propria produttività rompendo la barriera della distanza con il paziente fornendo un supporto indispensabile in caso di emergenze (telesoccorso), aumentando la tempestività alle esigenze diagnostiche (telediagnosi) e terapeutiche (teleassistenza); permette di monitorare con continuità un paziente piuttosto che effettuare unicamente controlli in ambulatorio in seguito ad episodi acuti e permette di archiviare una collezione di dati sulla storia clinica del paziente. Tutto questo enfatizza il ruolo centrale del paziente; per questo in letteratura si parla di "*person centric health system*".

Diverse soluzioni di telemonitoraggio sono presenti sul mercato, ognuna delle quali soddisfa particolari requisiti (riguardanti sicurezza, privacy, usabilità, robustezza, ..) ed è progettata ad hoc per realizzare determinate funzionalità. Analisi dello stato dell'arte dimostrano come tali sistemi siano efficienti nella riduzione dei costi della spesa sanitaria [4]. Nonostante ciò, una grande barriera ostacola ancora il percorso verso l'innovazione radicale nei sistemi sanitari finalizzata a ridurre ulteriormente i costi sanitari: la mancanza di interoperabilità nella rappresentazione della "conoscenza" sanitaria ("*interoperability at infor-*

mation level”) [5]. Portare l’interoperabilità nel servizio sanitario è una grande sfida che richiede innovazioni non solo nella tecnologia utilizzata nel sistema sanitario ma più in generale nel controllo e nello stile di lavoro [6].

Scopo di questa tesi è proporre al sistema sanitario un approccio all’interoperabilità basato sull’impiego di tecnologie semantiche e su un modello di condivisione dei dati concepito nell’ambito di un programma di ricerca finanziata europeo, e quindi di dimostrarne l’utilizzo in un caso specifico ancora non compiutamente esplorato, in cui vengono monitorate per essere esse in relazione informazioni originate da due ambiti diversi quello sanitario e quello ambientale: verranno dunque discusse l’architettura, il modello di rappresentazione dei dati e i principi ispiratori di una piattaforma interoperabile applicata al caso specifico del monitoraggio concorrente di parametri fisiologici ed ambientali. In letteratura sono infatti riconosciuti gli effetti negativi sul comportamento e sullo stato fisico di persone causati da fattori climatici quali caldo e umidità [7]. In questa sede viene fatta l’ipotesi che per poter comprendere e interpretare al meglio questi fenomeni, la conoscenza dei dati fisiologici e del contesto nel quale si trova il soggetto possa essere un fattore innovativo e utile per lo studio di correlazioni tra i parametri monitorati. Attualmente infatti gli studi di questi fenomeni, come l’onda di calore, richiedono molto tempo per essere investigati; spesso è molto difficile raccogliere dati e non sono considerati molti fattori come lo stato generale di salute del paziente e modifiche ai suoi dati fisiologici durante il fenomeno. Questa tesi offre una risposta a questa ipotesi investigativa. Recentemente sono stati sviluppati sistemi di monitoraggio portabili in grado di acquisire parametri fisiologici; alla luce di queste ricerche il nostro studio si basa sulla realizzazione di un sistema in grado di aggiungere anche informazioni sul contesto (attività svolta dall’utente, luogo in cui si trova, informazioni sull’ambiente). Saranno mostrati due approcci. Il primo approccio è un approccio di tipo classico: è stato sviluppato un sistema “chiuso” in grado di soddisfare le funzionalità richieste.

Il secondo approccio discusso ampiamente nella tesi è basato su un ambiente intelligente creato a partire da una piattaforma per l’interoperabilità (*Interoperability Platform*) sviluppata nell’ambito di un progetto europeo della JU ARTEMIS denominato SOFIA a cui l’autore ha partecipato.

Parte del lavoro descritto in questa tesi è stato pubblicato in [80] [81] [82][83][84][85][86][87][88].

2 La telemedicina: motivazioni e stato dell'arte

2.1 Cenni storici

In letteratura i primi riferimenti alla telemedicina risalgono agli anni settanta [8] ma in realtà la telemedicina ha preso piede velocemente solo dagli anni novanta cioè solo quando i servizi di comunicazione dati sono stati resi disponibili in larga scala sulle linee telefoniche [9]. In [10] è presentato il concetto di telemedicina ed è illustrata la storia della telemedicina mettendo in evidenza differenti applicazioni realizzate. L'Unione europea, nel 1990, ha concordato una definizione per la telemedicina stabilendo che essa comprende: *“Il controllo, il monitoraggio e la gestione dei pazienti, nonché la loro educazione e quella del personale, attraverso l'uso di sistemi che consentano un tempestivo accesso alla consulenza di esperti e alle informazioni del paziente, indipendentemente da dove risiedono”*.

L'OMS (Organizzazione Mondiale della Sanità) adotta nel 1997 la seguente definizione: *“la telemedicina è l'erogazione di servizi sanitari, quando la distanza è un fattore critico, per cui è necessario usare, da parte degli operatori, le tecnologie dell'informazione e delle telecomunicazioni al fine di scambiare informazioni utili alla diagnosi, al trattamento ed alla prevenzione delle malattie e per garantire un'informazione continua agli erogatori di prestazioni sanitarie e supportare la ricerca e la valutazione della cura”*. E' implicito in tale definizione il concetto per cui, quando si parla di telemedicina essa non deve essere confinata al singolo aspetto disciplinare introdotto, bensì a quell'insieme di applicazioni pratiche che oggi sono già in fase di implementazione.

Da allora un ampio numero di sistemi è stato realizzato con lo scopo principale di ridurre il tempo che il personale medico deve spendere per il paziente garantendo al contempo un alto livello di qualità. Inizialmente i sistemi di telemedicina erano dedicati a specifiche applicazioni quali tele-cardiologia, tele-radiologia, tele-dialisi,.. Con l'avvento di nuove piattaforme, rese possibili dalla presenza di Internet, incluse piattaforme mobili e con lo sviluppo di *Location Based Service*, la collezione di dati acquisiti da differenti sensori (indossabili, stazionari, posizionati in casa piuttosto che in altri ambienti) è diventata una realtà. Lo sviluppo di nuovi sensori e attuatori hanno aperto al strada a nuovi scenari in campo sanitario e nuove opportunità per i servizi di healthcare attualmente investigate in progetti di ricerca.

Il progetto OLDES Older people's e-services at home [11] è un progetto europeo finalizzato a sviluppare e progettare una soluzione tecnologica per fornire tele-assistenza e tele-compagnia a persone anziane; il sistema sviluppato porta a casa dell'anziano una serie di strumenti medici per la raccolta di dati fisiologici (diversi in base al profilo dell'utente) e servizi per favorire la partecipazione ad attività sociali e per fornire un accesso diretto di comunicazione verso altri utenti, personale medico e assistenti sociali. Un progetto simile è AILISA [12], un'iniziativa francese che promuove una piattaforma sperimentale per valutare tecnologie di monitoraggio per persone anziane nelle loro case. Sempre per il monitoraggio di persone anziane è finalizzato il progetto UbiSense [13] che invece esplora un nuovo approccio di monitoraggio basato sull'osservazione dell'attività del paziente. Tutti questi progetti traggono vantaggio dai progressi dell'ICT; questi sistemi come molti altri presenti sul mercato consentono di fornire assistenza a distanza e collezionare dati in modo automatico ma non forniscono un flusso di decisioni automatiche.

Questa funzionalità è invece fornita in quelle che sono le più moderne soluzioni di telemedicina ovvero le "*Health Smart Homes*"; questi sistemi nascono con l'obiettivo di facilitare la vita degli anziani o disabili nelle loro case. Per Smart Home si intende un'abitazione che incorpora una rete di comunicazione che è in grado di connettere apparecchi elettronici e servizi e permette di controllarli e monitorarli da remoto [14]. In particolare le *Health Smart Homes* dovrebbero garantire una vita autonoma a quelle persone che altrimenti sarebbero collocate in istituti [15]. Queste soluzioni sono nate dalle ricerche nel campo degli ambienti intelligenti e cercano di adattare la tecnologia ai bisogni dell'utente basandosi su tre concetti principali: *ubiquitous*

computing, ubiquitous communication e intelligent user interface. I risultati e le principali ricerche in tale ambito possono essere trovate in [16], [17], [18] [19], [20], [21], [22].

Una caratteristica comune e limitante di tutte le soluzioni sopra menzionate è che ogni implementazione è basata su propri protocolli proprietari, piattaforme, store di informazioni e modelli di rappresentazione dei dati. L'informazione prodotta da un sistema rimane all'interno del sistema stesso; la conseguenza ovvia è una frammentazione lungo il ciclo di assistenza sanitaria. Questa frammentazione è non solo causa di un aumento generale dei costi ma anche una barriera per quanto riguarda l'innovazione di servizi sanitari.

Se i dati sullo stato di salute e dati ambientali raccolti da sistemi di monitoraggio ambientale fossero interoperabili, sarebbe possibile combinarli e generare una nuova conoscenza in modo tale da poter realizzare applicazioni innovative a beneficio di differenti istituzioni e utenti.

2.2 L'interoperabilità

I sistemi interoperabili hanno un ruolo centrale in questa tesi. Qui saranno descritti i concetti di contesto, sistema interoperabile, ontologia e Smart Space.

2.2.1 Contesto

Un punto chiave di tutti i sistemi intelligenti è l'informazione sul contesto e come l'informazione è raccolta, scambiata e resa disponibile. Prima di illustrare la definizione di contesto nelle tecnologie dell'informazione, è utile partire dall'utilizzo del contesto nell'interazione umana. In accordo con Winograd [23] che presenta il concetto di contesto nel linguaggio, usiamo continuamente il contesto nella conversazione di tutti i giorni. Considerando la frase "Per favore sposta quello", la frase ha un significato ma letteralmente viene chiesto di spostare "quello"; senza conoscere a cosa si riferisce la conversazione, un osservatore non è in grado di capire cosa deve essere spostato. Conoscendo che i due interlocutori stanno parlando di un tavolo la frase assume il significato di "Per favore sposta il tavolo".

Il contesto in ambito delle tecnologie dell'informazione è presente da diversi anni variando dal log-in di un utente in un'applicazione al menu offerto dal right-click del mouse (menu contestuale). Molti autori hanno cercato di de-

finire il contesto negli anni alcuni dei quali hanno incluso dettagli per specifiche applicazioni. Una delle prime definizioni è la definizione del contesto di Schilit [24] che definisce il contesto come “dove sei, con chi sei e quali sono le risorse nelle tue vicinanze”.

Successivamente Chen e Kotz [25] hanno proposto la loro definizione: il contesto è un insieme di stati e configurazioni ambientali che insieme determinano il comportamento dell'applicazione o nei quali si verifica un evento che è interessante per l'utente.

Una più generale definizione appartiene a Dey e Abowd [26]. Il contesto è qualsiasi informazione che può essere usata per caratterizzare un'entità. Un'entità è una persona, un luogo o un oggetto che è considerato rilevante per l'interazione tra utente e applicazione, incluso l'utente e l'applicazione stessa.

Questa definizione di contesto è ampiamente accettata nel campo e sarà la definizione guida per questa tesi. In questa definizione di contesto è presente un tema ricorrente; la disponibilità di informazioni extra riguardanti lo stato dell'utente quali per esempio la sua posizione, l'identità, l'attività, il tempo,.. Per avere queste informazioni disponibili all'interno di un sistema, un utente deve essere equipaggiato con un certo numero di sensori per minimizzare l'intervento dell'utente stesso. I sensori giocano un ruolo fondamentale nella conoscenza del contesto, come affermato da Dey e Abowd. In questa tesi consideriamo un sensore come un elemento in grado di effettuare con ripetibilità misure di qualche fenomeno fisico e rappresentare il dato in un formato digitale per poi essere elaborato da un elaboratore. Sensori possono essere posizionati ovunque nell'ambiente; possono essere sensori personali (propri del dispositivo dell'utente) o sparsi nell'ambiente. Ovviamente raccogliere dati da un sensore non è sufficiente per avere una ricostruzione della conoscenza del contesto; per produrre informazioni contestuali di più alto livello (per esempio risalire all'attività dell'utente) è necessario combinare i dati raccolti da più sensori con tecniche di sensor fusion e algoritmi di machine learning.

Il contesto è utilizzato in differenti sistemi; alcuni di essi utilizzano le informazioni contestuali per cambiare il proprio comportamento mentre altre si limitano a visualizzare il contesto corrente. In letteratura esistono diverse definizioni di context-awareness e revisioni possono essere trovate in [27] e in [28]. Dey e Abowd propongono la seguente definizione: un sistema è context-aware se utilizza il contesto per fornire informazioni rilevanti e/o servizi all'utente, dove la rilevanza viene a dipendere dagli obiettivi dell'utente.

2.2.2 Sistema interoperabile

La complessità nella progettazione di sistemi di tipo context-aware riguarda settori di ricerca che coinvolgono differenti domini. Infatti questi sistemi richiedono una conoscenza di elementi eterogenei che vanno dai sensori alle architetture hardware, dai sistemi di comunicazione ai modelli di rappresentazione dei dati. Le ricerche correnti infatti si occupano di task specifici che includono sensori, middleware e lo sviluppo di applicazioni in grado di utilizzare informazioni contestuali. Per realizzare un buon sistema context-aware la visione globale dell'architettura è il punto di partenza considerando le specifiche e i requisiti. Un aspetto chiave ma spesso ignorato è l'interoperabilità delle informazioni contestuali. Il tema dell'interoperabilità è stato investigato per anni; l'interoperabilità tra sistemi eterogenei e distribuiti in grado di raccogliere informazioni dallo spazio fisico è stata infatti investigata sempre più in seguito all'evidente realizzazione della visione di Weiser sul pervasive computing [29]. In [30] si possono trovare differenti definizioni ma l'IEEE ha definito l'interoperabilità come l'abilità di due o più sistemi o componenti di scambiarsi e utilizzare informazione [31].

Una più ampia definizione proposta da Brownsword [30] si avvicina meglio all'uso del concetto di interoperabilità trattata in questa tesi. Brownsword definisce l'interoperabilità come l'abilità di un insieme di entità in grado di comunicare di condividere specifiche informazioni e operare su queste informazioni in accordo con una determinata semantica. Questa definizione combacia con il nostro scopo dove le entità in grado di comunicare sono dispositivi elettronici e persone e non è specificato un determinato tipo di comunicazione. Un altro punto chiave di questa definizione che sarà fondamentale in questa tesi è l'utilizzo di una semantica, cioè il significato dell'informazione.

In questo lavoro basandoci su [32] sarà considerata una netta distinzione tra i seguenti livelli concettuali di interoperabilità: *communication level*, *service level*, *information level*.

- *Information level* : per *information (o semantic) interoperability* si intende la condivisa comprensione del significato dell'informazione;

- *Service level* : la *service interoperability* è l'abilità di più sistemi di condividere, scoprire e comporre servizi;
- *Communication level*: la *communication level* è l'interoperabilità a livello OSI 1,2,3 e 4 [33] .

Questa tesi si focalizzerà principalmente sull'interoperabilità semantica. Un ramo dell'intelligenza artificiale chiamato rappresentazione dell'informazione studia questa problematica da anni. Una parte dell'informazione chiamata ontologia è l'elemento più all'avanguardia dell'ultimo decennio in questo campo.

2.2.3 Ontologia

Il concetto di ontologia in informatica nasce dagli studi filosofici su ciò che esiste [34]. Questi studi includono la concettualizzazione delle relazioni tra idee, eventi e entità. In informatica, in particolare nel campo dell'intelligenza artificiale, il concetto di ontologia è stato introdotto per avere un linguaggio formale per la descrizione dei concetti all'interno di un determinato dominio e una descrizione di come sono collegati fra loro. Gruber [35] definisce un'ontologia come un'esplicita descrizione di una concettualizzazione; per concettualizzazione Gruber intende una visione astratta e semplificata del mondo che si vuole rappresentare per determinati scopi. In questa tesi il concetto di ontologia si riferisce ad un'esplicita definizione della conoscenza di una determinata area di interesse (dominio) che è rappresentata in maniera organizzata. Una dettagliata descrizione sulla storia dell'ontologia si può trovare in [36]. Le ontologie sono utilizzate da persone, database e applicazioni che necessitano di condividere la rappresentazione dell'informazione appartenente ad un determinato dominio [37]. Le ontologie sono un caposaldo dell'interoperabilità in quanto forniscono un modo per applicazioni eterogenee di condividere informazioni basate su una semantica stabilita. Senza un'ontologia o una concettualizzazione della conoscenza che viene rappresentata non è possibile avere un vocabolario per la rappresentazione della conoscenza [38].

Per descrivere i concetti di un dominio un'ontologia utilizza i concetti di classe. Un esempio classico illustrato in [39] è la rappresentazione della classe *vino*. Questa classe rappresenta tutti i vini disponibili nell'ontologia. Le ontologie utilizzano il concetto di eredità e permettono la creazione di sottoclassi.

Un esempio di questo è l'aggiunta della sottoclasse *rosso* e a sua volta della sottoclasse *Montefalco*. Produzioni specifiche di Montefalco potrebbero essere istanze o individui della classe *Montefalco*. Le relazioni tra le classi sono rappresentate definendo le proprietà. Una proprietà potrebbe essere *intensità* che consente di connettere il concetto di un particolare istanza di vino rosso con un'intensità come per esempio *deciso*.

L'information interoperability in questa tesi è basata pesantemente sulla rappresentazione della conoscenza condivisa che un'ontologia è in grado di fornire.

2.2.4 Smart Space

Prima di definire uno Smart Space (SS) è necessario definire uno Smart Environment. Cook [40] definisce uno Smart Environment un sistema in grado di acquisire e applicare conoscenza su un ambiente e capace di adattarsi ai propri abitanti in modo tale da migliorare la loro esperienza in quell'ambiente. In questa tesi lo Smart Space è una porzione fondamentale dello Smart Environment. Per SS intendiamo una rappresentazione digitale dell'informazione rappresentante il modo fisico e tale da offrire un'estensione per la ricerca dell'informazione attraverso un Semantic Information Broker (SIB).

La SIB è un'entità (posta sull'*information level*) per la memorizzazione, la condivisione e la gestione dell'informazione dello SS.

In questa tesi l'informazione è rappresentata nella SIB mediante un grafo che può essere scritto e/o letto; l'informazione del grafo è rappresentata mediante un'ontologia per fornire univocamente significato all'informazione.

2.3 Struttura della tesi e contributi

Nel capitolo 3 è presentato lo sviluppo ed il progetto di un sistema indossabile realizzato per contestualizzare il segnale elettrocardiografico; si tratta di una soluzione classica. Nel capitolo 4 sono invece descritte le tecnologie del web semantico che sono fondamentali per realizzare sistemi interoperabili; gli Spazi Intelligenti e la piattaforma di supporto utilizzata sono illustrati nel capi-

tolo 5. Il capitolo 6 descrive una metodologia per realizzare un'ontologia e l'ontologia stessa usata nell'applicazione del capitolo 7; in quest'ultimo capitolo è descritto il prototipo di Smart Space per il monitoraggio concorrente di parametri fisiologici e ambientali.

Infine nel capitolo 8 sono riportate le conclusioni contenenti una serie di raccomandazioni per lo sviluppo di Spazi Intelligenti e gli sviluppi futuri.

Progettisti di sistemi a basso livello considerano il loro sistema isolato, senza tenere conto di tutti i possibili usi dei dati che saranno generati. Se il progettista considerasse l'utilizzo in un sistema a più alto livello, sistemi interoperabili potrebbero essere realizzati con minor sforzo e maggior velocità.

I sistemi interoperabili offrono molti vantaggi; il principale di questi è l'abilità di condividere i dati tra differenti sistemi. La riusabilità dei dati permette lo sviluppo spontaneo di applicazioni nella forma di mashups ovvero di applicazioni in grado di utilizzare parti di codice già esistente e fonderle per fornire nuove funzionalità all'utente finale. Al tempo stesso quando nuovi sensori e sistemi sono aggiunti i servizi già esistenti sono in grado di continuare a funzionare correttamente e alcuni di essi di trarre beneficio dai nuovi dati senza dover modificare le applicazioni già esistenti. La flessibilità è una caratteristica fondamentale e coerente con la visione originale di Weiser dovendo un sistema interoperabile tenere conto di diversi tipi di integrazione.

Nell'approccio per il design di un sistema interoperabile, il progettista può usare una separazione iniziale disaccoppiando hardware e software in modo intelligente. In questo modo software e hardware a basso livello possono essere progettati in modo rapido e in modo convenzionale e realizzando su di essi le applicazioni innovative.

Una soluzione basata su SS astrae le complicazioni da parte degli sviluppatori di dover definire un dizionario standard per la condivisione dell'informazione. Questo semplifica l'integrazione di nuovi dispositivi di origine eterogenea durante l'evoluzione del sistema.

L'obiettivo principale della tesi è quello di analizzare il progetto di un sistema interoperabile dando raccomandazioni per la realizzazione di una soluzione di tale tipo.

Due differenti tipo di progettazione sono stati scelti per la discussione: il primo utilizza un sistema integrato, utilizzando sensori esistenti ai quali è stato aggiunto un'interfaccia per l'acquisizione del segnale ECG. Il secondo implementa un sistema di monitoraggio basato sul concetto di SS.

3 Progetto e sviluppo di un sistema indossabile per la contestualizzazione del segnale ECG

In questo capitolo è illustrato il progetto e la realizzazione di un sistema integrato e portatile in grado di monitorare informazioni fisiologiche e il contesto dell'utente; in questo scenario per contesto intendiamo la situazione globale in cui si trova l'utente (dal tipo di attività che svolge, al luogo in cui si trova, alle informazioni sull'ambiente circostante). Nella fase di progettazione sono stati considerati differenti fattori dalla disponibilità hardware alla possibilità di poter evolvere il prototipo. Lo schema a blocchi del sistema è illustrato in figura 1.

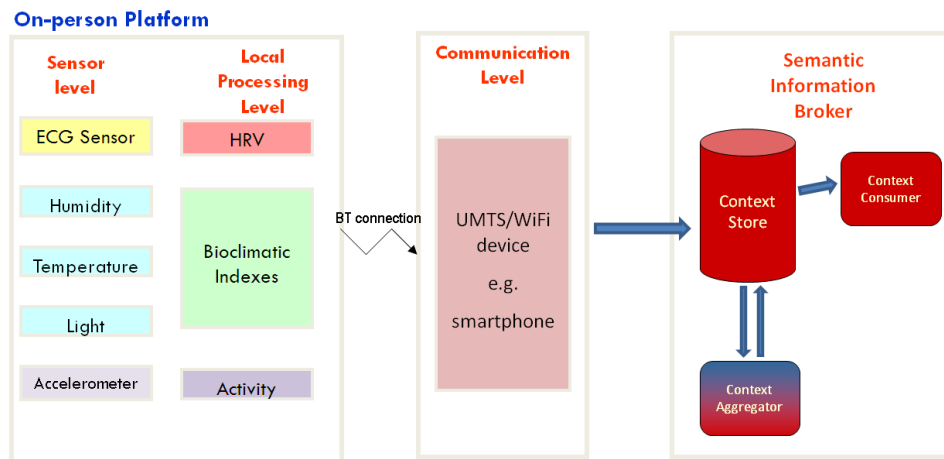


Figura 1 Visione del sistema di monitoraggio

Al livello più basso, sensori su dispositivi portabili raccolgono informazioni; questi dati “grezzi” sono elaborati localmente ed aggregati per estrarre informazioni di più alto livello per poi essere consegnate in un formato standardizzato per le infrastrutture a più alto livello; qui si collocano le infrastrutture di *context management* consentendo di raccogliere dati anche da

altre fonti e di fornire gli stessi dati in modo standardizzato ad applicazioni e servizi.

In particolare in questo capitolo viene descritto lo sviluppo del prototipo portatile per il monitoraggio rappresentato dal blocco “*On-person platform*” nella figura 1; l’obiettivo è quella di avere un device flessibile in grado di acquisire segnali da vari sensori per poter raccogliere informazioni sullo stato di salute, sull’attività della persona e sull’ambiente in cui si trova il soggetto; un punto chiave è la capacità del device di poter subire upgrade aggiungendo o cambiando sensori.

Considerando il tracciato ECG il segnale principe dei segnali biomedici, ci siamo concentrati su questo dato e sulle informazioni del contesto.

Analizzando le tecnologie attuali, lo stato dell’arte e le nostre necessità, abbiamo deciso di utilizzare un nodo di una rete di sensori (Wireless Sensor Network) per implementare l’unità base portatile del nostro sistema. Il termine WSN indica una rete di sensori senza fili, composta da un certo numero di dispositivi distribuiti nello spazio per monitorare contemporaneamente condizioni fisiche o ambientali, come temperatura, luce, movimento. I nodi di queste reti, detti *motest* (abbreviazione di “remote”, in inglese) sono piccoli computer autonomi, alimentati a batteria e dotati di un certo numero di sensori e di un collegamento radio (o altri mezzi di comunicazione senza fili). Originariamente nati per applicazioni militari, i sistemi WSN sono oggi impiegati per un crescente numero di applicazioni, come l’agricoltura, il monitoraggio di strutture e terremoti, controlli industriali, automazione domestica, applicazioni sanitarie.

La scelta è motivata dalla flessibilità offerte dal singolo nodo, dalle ridotte dimensioni, dalle prestazioni offerte e dalla possibilità intrinseca di poter realizzare una Body Area Network.

3.1 Architettura del sistema

Il prototipo permetterà il monitoraggio di diverse grandezze acquisite da differenti sensori tramite iMote2 e trasmetterà i dati ad un dispositivo mobile attraverso Bluetooth. La scelta di utilizzare uno smartphone è motivata dal fatto che è un dispositivo personale che permette all’utente di accedere al servizio ovunque. Questo dispositivo mobile trasmette i dati raccolti usando la connes-

sione al momento disponibile (o UMTS o 802.11) verso il repository di sto.

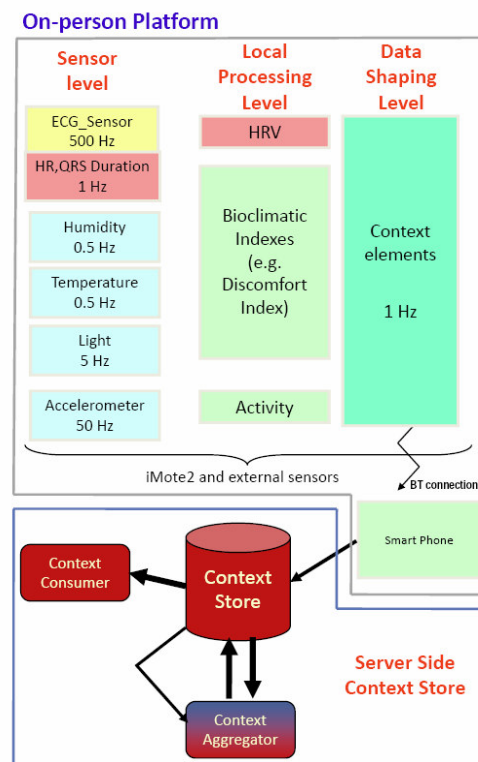


Figura 2 Architettura del sistema

3.2 Una piattaforma multi sensore: l' iMote2

Gli iMote2 sono stati sviluppati dai laboratori Intel e attualmente sono sul mercato grazie la Crossbow Technology. Sono nodi ad alte performance sviluppati per la ricerca in ambito di Wireless Sensor Network. Gli iMote2 sono i dispositivi più performanti tra quelli sviluppati nel campo delle WSN e risulta netta la differenza con altri tipi di motes. Normalmente infatti questi dispositivi sono basati su microprocessori a 8/16 bit con relativamente poco spazio disponibile per il codice; un confronto con altre piattaforme è presente in [41]. Gli iMote2 sono basati su un processore a 32 bit PXA271 ARM e un ChipCon. Più dettagli sulla piattaforma possono essere trovati in [42].

Il dispositivo iMote2 presenta una struttura modulare essendo composto di diverse boards interconnesse attraverso connettori posti su entrambi i layers delle schede . Le schede fornite da Intel sono:

- processor board: include il microprocessore PXA271 che integra 256KB di SRAM, 32 MB di FLASH e 32 MB di SDRAM, ed il modulo radio con relativa antenna; il processore integra alcuni set di I/O rendendo la piattaforma estremamente dinamica supportando diversi standard di comunicazioni (I2C, SPI, UART);
- sensor board: questa scheda integra vari sensori tra cui sensore di luce (TAOS TSL2561 [43]), un sensore di temperatura e umidità (Sensirion SHT15 [44]), un ulteriore sensore di temperatura (Ttmp175), un accelerometro 3d (ST Micro LIS3L02DQ [45]) ed infine un convertitore analogico digitale (Maxim MAX1363 4);
- debug board: consente di installare il software nella memoria flash del microcontrollore e consente il debug attraverso l'interfaccia JTAG;
- battery board: permette di alimentare il mote qualora non sia connesso tramite USB al PC.

La Imote2 sensor board, sviluppata dalla stessa Intel, fornisce diversi sensori; infatti sono presenti sulla board un sensore di temperatura, umidità, luce, un accelerometro triassiale e un convertitore analogico digitale. La temperatura e l'umidità sono forniti dal Sensirion SHT15 ; la massima frequenza di campionamento del sensore è di 1 Hz e la risoluzione è di 12 bits per l'umidità relativa e di 14 bits per la temperatura. Il sensore di luce è il TAOS TSL2561; il sensore ha una risoluzione di 16 bit e la frequenza di campionamento è impostabile dall'utente. L'accelerometro triassiale è il ST Micro LISL02DQ; consente la lettura dei dati per i 3 assi con una risoluzione di 12 bit. L'ADC è il MAX1363 con una risoluzione a 12 bit.

L' iMote2 pertanto è un'ideale punto di partenza offrendo un dispositivo portatile dotato di sensori utili per la raccolta di informazione sull'attività e sull'ambiente e di una unità ad elevate prestazioni per implementare del local processing.

Quindi abbiamo avuto l'esigenza di realizzare due ulteriori schede da "impilare" nel sistema iMote2; una scheda per integrare la comunicazione con protocollo Bluetooth, standard consolidato in grado di permetterci di comunicare wireless con dispositivi portabili o PC, ed una scheda per l'acquisizione del segnale ECG.

3.2.1 Sistema operativo

Il primo passo è stato analizzare la piattaforma iMote2 dal punto di vista software e hardware. Al momento dello sviluppo tre differenti sistemi operativi erano disponibili per la piattaforma fornendo diversi livelli di supporto, mostrati in tabella.

OS	Supporto Hardware	Astrazione Hardware	Livello di documentazione
TinyOS 1.x	Eccellente	Scarsa	Scarso
TinyOS 2.x	Buona	Buona	Eccellente
Linux	Buona	Eccellente	Scarso

Diversi criteri sono stati presi in considerazione per la scelta del sistema operativo per sviluppare il nostro sistema. La quantità di supporto hardware disponibile per ognuno dei diversi OS è un aspetto critico in quanto lo sviluppo di un nuovo driver richiede molto tempo. Quindi un altro aspetto era l'astrazione fornita dal sistema operativo cioè la facilità offerta per scrivere programmi avendo a disposizione i driver. Infine un altro aspetto critico è stato la valutazione della documentazione disponibile. La documentazione che spiega il funzionamento di un sistema operativo dedicato ad una particolare piattaforma è fondamentale. Infatti le release disponibili per iMote non erano molto documentate e spesso erano presenti molte modifiche effettuate dagli sviluppatori. La nostra scelta è ricaduta sul sistema operativo più nuovo, TinyOS2.x.

Alcuni aspetti chiave da considerare quando si utilizzano WSN sono l'interazione con l'ambiente attraverso i sensori, la comunicazione e l'alimentazione senza fili e la limitatezza delle risorse disponibili. In genere, sebbene siano vincolati da queste limitazioni, i motes devono essere molto reattivi e partecipare all'esecuzione di complessi algoritmi ad esempio per l'aggregazione di dati e/o la localizzazione. Questa combinazione di requisiti rende i tradizionali sistemi operativi inappropriati. Pertanto, sono stati realizzati sistemi operativi specifici per nodi sensore, tra cui TinyOs che si prefigge di venire incontro a queste esigenze. TinyOS2 è un sistema operativo open-source sviluppato appositamente per le reti di sensori dall'Università di Berkley e il centro di ricerche Intel. Esso si differenzia in modo sostanziale dai sistemi ope-

rativi general-purpose, integrandosi direttamente con le applicazioni sviluppate; una tipica applicazione TinyOS based ha le dimensioni di una decina di KB, dei quali soltanto 400 byte competono al sistema operativo; è proprio alle dimensioni esasperatamente ridotte che TinyOS deve il suo nome (“*TinyOS is a tiny operating system*”). Esso presenta le seguenti caratteristiche: architettura basata su componenti, modello di concorrenza basato su eventi e task, operazioni ‘split-phase’.

Scelto il sistema operativo è stato necessario configurare e capire la procedura necessaria per sviluppare con la combinazione software/hardware. Il supporto è stato rilasciato da Intel. Il primo passo è consistito nel testare i sensori; per fare ciò i dati grezzi dei sensori sono stati spediti direttamente su seriale per avere subito un feedback del corretto funzionamento nel terminale. Testati i sensori e presa confidenza con la piattaforma, il passo successivo è consistito nel risolvere il problema dell’ acquisizione del segnale ECG e della comunicazione. Infatti per permettere al sistema di poter essere utilizzato ovunque quest’ultimo deve poter comunicare con la rete ovunque. L’iMote2 non fornisce questo supporto per cui è stato utilizzato uno smartphone Windows Mobile come ponte. Per beneficiare dei servizi che fornisce uno smartphone, è stato necessario integrare una connessione Bluetooth alla piattaforma iMote2. Quindi abbiamo avuto l’esigenza di realizzare due ulteriori schede da “impilare” nel sistema; una scheda per permettere la comunicazione con protocollo Bluetooth ed una scheda per l’acquisizione del segnale ECG.

3.3 Acquisizione del segnale ECG

Per realizzare un prototipo portatile in grado di monitorare informazioni sul contesto e il segnale ECG quale principale parametro fisiologico, si è deciso di realizzare una scheda da impilare alla struttura modulare dei motes per l’acquisizione e l’elaborazione del segnale ECG.

Il progetto della scheda ha richiesto alcune considerazioni; la complessità del sistema infatti è legata alla modalità di utilizzo (a riposo o in movimento), al numero di derivazioni (cioè al numero di segnali acquisiti contemporaneamente) e alla struttura software/hardware utilizzate per filtrare/elaborare il segnale. Il nostro requisito consiste nell’ avere un sistema in grado di monitora-

re il paziente durante le proprie attività giornaliere. Il progetto di un sistema di monitoraggio holter di questo tipo richiede una frequenza di campionamento di circa 500 Hz per applicazioni che analizzano la heart rate variability e la variabilità dell'intervallo PR [46].

La scheda da noi realizzata è composta da due stadi: un front-end analogico per l'acquisizione del segnale ECG e una sezione digitale per il campionamento, la elaborazione e la comunicazione dei dati.

3.3.1 Front-end analogico

Lo schema a blocchi della sezione analogica è mostrato in figura 3.

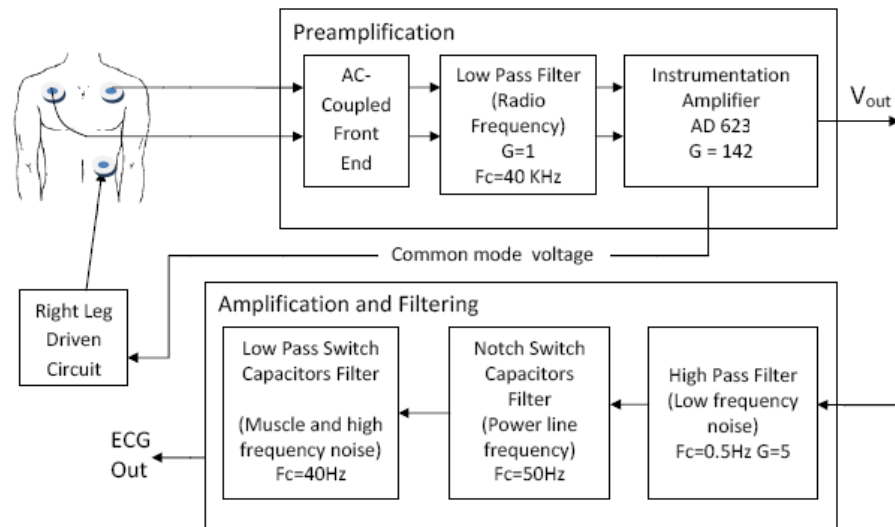


Figura 3 Schema a blocchi della sezione analogica della scheda ECG

E' stato scelto di realizzare un circuito di acquisizione ad una derivazione con tre elettrodi. Per ridurre gli effetti dovuti alle capacità parassite tra i cavi e la massa, la massa dei cavi è guidata dalla tensione di modo comune. Il primo stadio della catena di preamplificazione è un *AC-coupled front-end* [47] implementato da una rete passiva che fornisce un elevato CMRR; questa configurazione è stata scelta per aumentare l'immunità ai disturbi dovuti a effetti capacitivi e per diminuire gli sbilanciamenti nello stato differenziale di ingresso. Quindi un filtro passivo passa basso ad alta frequenza di taglio è stato usato per abbattere le interferenze RF. Il segnale differenziale è quindi amplificato da un amplificatore da strumentazione caratterizzato da bassi consumi e da alti valori di CMRR; l'amplificatore usato è l'AD623 della Analog Devices. Il segnale è quindi filtrato da un filtro passivo passa alto con frequenza di taglio

di 0.5 Hz e amplificato in modo tale da poter sfruttare l'intero range del convertitore A/D. A valle il segnale passa attraverso due filtri a capacità commutate del 5 ordine; i filtri utilizzati sono prodotti dalla Maxim. Un MAX 7414 è stato configurato per realizzare un filtro notch per rimuovere i disturbi della frequenza della rete. Infine un MAX 7413 nella sua configurazione standard realizza un filtro passa basso di Bessel con frequenza di taglio a 40 Hz. La scelta dei filtri a capacità commutate è stata motivata dalla accuratezza fornita, dal fatto che richiedo pochi componenti esterni per essere implementati e dal comportamento noto del filtro nelle condizioni di lavoro utilizzate.

3.3.2 Sezione digitale

A seguito dello stadio di acquisizione analogico, un MSP430 si occupa di digitalizzare, filtrare ed elaborare il segnale ECG. Questo microcontrollore caratterizzato da bassi consumi e da un elevato numero di periferiche integrate è risultato ideale per questa applicazione. Infatti ha permesso di usufruire dell'ADC a 12 bit integrato, del moltiplicatore hardware e della comunicazione UART per comunicare con il mote. L'MSP infatti è utilizzato per effettuare del local processing per estrarre l'Heart Rate. Il riconoscimento del complesso QRS in real time è stato realizzato utilizzando una variante del ben noto algoritmo di Pan e Tompkins [48]. L'algoritmo usato sfrutta i passi standard dell'algoritmo di Pan e Tompkins e invece di utilizzare la soglia adattiva per il rilevamento del complesso, il nostro algoritmo usa una soglia simile a quella proposta in [49].

Quindi per evidenziare il complesso QRS sono stati realizzati a livello di firmware 2 filtri FIR a 17 taps; un filtro passa banda con banda passante compresa tra 6 Hz e 30 Hz e un filtro passa alto con frequenza di taglio di 2 Hz. Il segnale filtrato è differenziato (tramite un FIR a 5 taps), elevato al quadrato e infine passato ad un integratore a media mobile di 150 ms. Come passo finale una soglia adattiva è applicata per il rilevamento del complesso QRS. La soglia è la composizione di due parametri indipendenti: uno basato sulla derivata del segnale e uno sul tempo previsto per il verificarsi del battito successivo.

3.4 Considerazioni

Grazie alla struttura modulare, abbiamo potenzialmente ottenuto il sistema portabile (processor board, sensor board, ECG board, Bluetooth board, battery board). In figura 4 è mostrato il flusso dei dati dai sensori al modulo di trasmissione Bluetooth evidenziando i canali di comunicazione tra le varie schede.

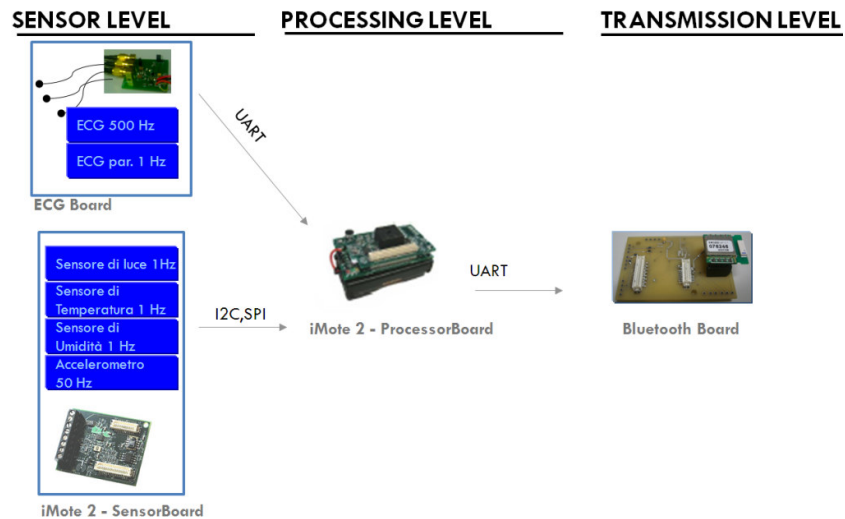


Figura 4 Flusso dei dati nella piattaforma indossabile

Il sistema operativo TinyOS è scritto nel linguaggio di programmazione nesC: un'estensione del linguaggio C studiata appositamente per applicazioni WSN e ottimizzato per scrivere codice strutturato basato sui componenti. Tramite questo linguaggio si è realizzata un'applicazione in grado di acquisire i dati dei sensori della sensor board, di ricevere i dati dalla scheda ECG, di effettuare del local processing (per ora limitato solamente al calcolo di indici ambientali e al calcolo di lux), di trasmettere i dati tramite Bluetooth. La scheda ECG può inviare i pacchetti con i dati elaborati (HR medio, singoli intervalli RR) e pacchetti con il segnale ECG grezzo. Il mote invia regolarmente con frequenza di 1 Hz un pacchetto dati contenente informazioni riassuntive (HR medio, intervalli RR, accelerazioni sui 3 assi, temperatura, umidità, indice bioclimatico di Thom [50], quantità di luce) ed eventualmente in continua il tracciato del segnale ECG; inoltre invia ogni minuto un ulteriore pacchetto contenente un timestamp prelevato dal real time clock del PXA271.

I dati inviati dal prototipo sono riportati in tabella.

	Periodo (s)	Valore	Tipo di dato
Heart Rate	1	16 bit Int	Dato fisiologico
Intervalli RR	1	Nd 16 bit Int	Dato fisiologico
Intensità di luce	1	16 bit Int	Dato ambientale
Temperatura	1	16 bit Int	Dato ambientale
Umidità	1	16 bit Int	Dato ambientale
Thom Index	1	16 bit Int	Dato ambientale
Dati accelerometro	1	3d 16 bit Int	Dato fisiologico
Timestamp	60	2d 16 bit Int	Temporizzazione
ECG segnale grezzo (opzionale)	Trasmissione in continua	16 bit Int a campione	Dato fisiologico

La ridondanza di filtri (analogici e digitali) nel prototipo della ECG board è stata conseguenza del fatto di voler effettuare stime di consumi e prestazioni per poter di conseguenza implementare la soluzione ottimale.

Per quanto riguarda la gestione dei consumi il punto critico risulta essere la piattaforma iMote e la scheda per la trasmissione Bluetooth basata su il modulo di trasmissione Parani esd200. Volendo ridurre i consumi, sono state sfruttate le modalità a basso consumo dell'MSP430. L'MSP è normalmente configurato nello stato di Low Power Mode 3 (in cui consuma 0,5 mW) e commuta in Active Mode ad ogni interrupt dell'ADC per effettuare l'elaborazione locale; campionando con una frequenza di 500 Hz il profiling ci ha mostrato che l'MSP è in LPM3 per l'80% del tempo, portando ad un consumo medio (parte analogica e digitale della scheda ECG) di 16 mW.

In tabella i dettagli dei consumi del sistema.

	Corrente (mA)	Tensione (V)	Potenza mW
Analog ECG board	4,7	3	14,1
MSP430 LPM3	0,16	3,3	0,53
MSP430 Active	3	3,3	9,9
iMote2 + SensorBoard	38	4,5	171
Bluetooth Board	47	3,28	154,16

Un problema riscontrato è il posizionamento del sensore di luce; infatti normalmente l'utente indossa il sistema posizionando il mote in cintura; in questo modo l'angolo di incidenza della luce ambientale sul sensore non è abbastanza ampio da permettere di avere una misura accurata e consistente

4 Le tecnologie per l'interoperabilità delle informazioni

In questo capitolo saranno illustrati nel dettaglio i concetti di ontologia e di Smart Space necessari per comprendere il funzionamento di un sistema interoperabile.

4.1 Ontologia

Nell'ultimo decennio il W3C [51] ha introdotto novità per quanto riguarda le specifiche per il web semantico. L'obiettivo principale del web semantico è quello di rendere l'informazione leggibile dalla macchina, disponibile per la condivisione e il riutilizzo. Al centro della visione della W3C sono presenti le ontologie finalizzate a rendere l'informazione interoperabile per permettere alle macchine di condividere una comune base di informazioni. Le tecnologie utilizzate dal W3C possono essere pensate come un sistema diviso in livelli come mostrato in figura 5. Il W3C ha adottato il Resource Description Framework (RDF) come modello di rappresentazione dei dati; su tale modello sono sviluppati RDF Schema (RDFS) e il Web Ontology Language (OWL).

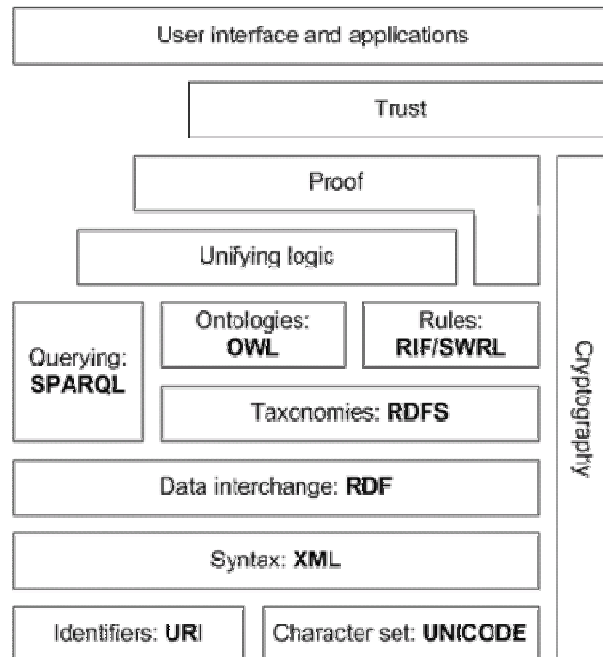


Figura 5 W3C Semantic Web Stack

4.2 RDF e RDFS

L’RDF è un linguaggio originariamente creato dal W3C come modello di rappresentazione di metadati [52]. Lo scopo originale del linguaggio era descrivere le risorse disponibili sul Web. Il suo utilizzo si è poi evoluto dalla descrizione delle risorse Web a metodi più generali per modellare concettualmente l’informazione.

Il modello di dati presentato dall’RDF è basato sulla descrizione di risorse nella forma di espressioni basate su triple del tipo soggetto – predicato – oggetto. Il soggetto è una risorsa, il predicato (o proprietà) esprime una relazione tra il soggetto e l’oggetto. L’ RDF fornisce una metodologia robusta per esprimere la semantica dell’informazione identificando univocamente le risorse tramite Uniform Resource Identifiers (URI). Un URI fornisce un identificatore unico per una risorsa. Un esempio di tripla :

- Soggetto: *http://www.mml. arces.unibo.it/fvergari#Person0001*
- Predicato: *http://www.mml. arces.unibo.it/fvergari#HasName*
- Oggetto: *“Fabio Vergari”*

Quando differenti e separati sistemi iniziano ad utilizzare gli stessi URI per descrivere l'informazione, il significato semantico delle risorse è condiviso fra diverse applicazioni. Un insieme di triple RDF rappresenta un grafo come mostrato in figura 6.

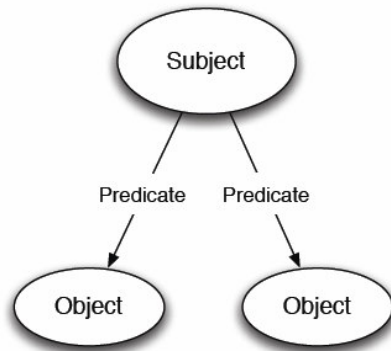


Figura 6 Esempio di grafo RDF

I soggetti rappresentano i nodi di partenza, i predicati gli archi tra i nodi e gli oggetti i nodi di arrivo. Un sottografo RDF rappresenta un insieme di triple contenuto nel grafo.

L'RDF può essere serializzato in diversi formati come l' XML-RDF [53], Notation3 [54], Turtle [55].

IL W3C ha introdotto inoltre l'RDFS per permettere la creazione di vocabolari RDF [56]. L'RDFS definisce classi e proprietà utili per la descrizione di classi, proprietà e altre risorse. Questo consente di generare tassonomie permettendo di creare relazioni di gerarchia tra le classi che saranno definite. Inoltre le proprietà vengono caratterizzate da dominio e range in modo tale da poter essere applicate a solo determinati tipi di risorse. Poiché l'OWL nasce direttamente utilizzando i concetti dell'RDFS, alcune notazioni sono qui presentate; la sintassi utilizzata è Notation3. NotationN3 utilizza le seguenti abbreviazioni.

a <<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>>
= <<http://www.w3.org/2002/07/owl#sameAs>>

Definizione di classe: per definire una classe viene utilizzata la proprietà *rdfs:class*.

```
@prefix default : <http://www.mml.arces.unibo.it/fvergari#> .
@prefix rdfs : <http://www.w3.org/2000/01/rdf-schewan#> .
@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
default : Person
    a      rdfs : Class .
```

Person è così definita una classe RDFS

Definizione di sottoclasse: per definire una sottoclasse viene utilizzata la proprietà *rdfs:subClassOf*.

```
@prefix default : <http://www.mml.arces.unibo.it/fvergari#> .
@prefix rdfs : <http://www.w3.org/2000/01/rdf-schewan#> .
@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
default : Person
    a      rdfs : Class .
default : Female
    a      rdfs : Class ;
    rdfs: subClassOf default: Person .
```

In questo modo *Female* è sottoclasse di *Person* ereditando tutte le proprietà e restrizioni applicate a *Person*.

Definizione di dominio e range di proprietà: L'RDFS permette di assegnare alle proprietà restrizioni per le classi dei soggetti e oggetti.

```
@prefix default : <http://www.mml.arces.unibo.it/fvergari#> .
@prefix rdfs : <http://www.w3.org/2000/01/rdf-schewan#> .
@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
default : Person
    a      rdfs : Class .
default : Human
    a      rdfs : Class ;
default : HasSpecies
    a      rdfs : ObjectProperty ;
    rdfs: domain default: Person ;
```

rdfs : range default : Human .

In questo modo è definita la proprietà *HasSpecies* che ha dominio *Person* e range *Human*.

4.3 OWL e RDF++

L'OWL è costruito sulle basi dell'RDF e dell'RDFS. E' un linguaggio utilizzato per descrivere il significato della terminologia utilizzata nell' RDF e nell'RDFS. L'OWL fornisce un vocabolario per descrivere proprietà e classi, relazioni tra classi, cardinalità, uguaglianza, tipi di proprietà, caratteristiche delle proprietà [57].

A metà degli anni cinquanta furono inventate le reti semantiche da Richard Richens e definite come un semplice modello di riferimento per supportare la concettualizzazione [58]. Nel 1970 Minsky sviluppò Frames, una struttura dati per rappresentare situazioni stereotipate [59], consentendo la creazione in modo più semplice di classi e istanze nel modello. Nella metà degli anni ottanta apparvero le logiche descrittive per rappresentare concetti, regole, individui e relazioni. Solo nel 2002 DAML + OIL fu sottomessa al W3C e utilizzata come punto di partenza per lo sviluppo dell'OWL.

L'OWL è utilizzato per formalizzare un dominio definendo le classi e le proprietà tra queste, definire gli individui e le asserire proprietà tra essi, e effettuare reasoning sulle classi e sugli individui al grado permesso dalla semantica fornita dal linguaggio OWL [60]. Un reasoner è una parte di codice in grado di dedurre conseguenze logiche dagli assiomi espressi nell'ontologia.

L'OWL è divisa in tre sottolinguaggi, ognuno dei quali è un'estensione del predecessore: OWL Lite, OWL DL e OWL Full.

- *OWL Lite* è il più semplice tra i linguaggi OWL offrendo vantaggi prestazionali; la cardinalità supportata permette solo i valori 0 o 1.
- *OWL DL* o *OWL Description Logics* supporta il massimo dell'espressività fornendo buone prestazioni computazionali. Ha una forte corrispondenza con la Description Logics e fornisce restrizioni utilizzate per fornire supporto a reasoner.
- *OWL Full* fornisce il massimo dell'espressività fornendo la libertà dell'RDF. Espressività e libertà incidono negativamente al supporto

per reasoner; non è garantito che siano calcolabili tutte le deduzioni possibili.

Nell'ambito della tesi sarà utilizzato un piccolo sottoinsieme dell'OWL Lite, chiamato RDF++ [61]. I costrutti supportati dall'RDF++ oltre a quelli forniti dall'RDFS comprendono anche *owl:sameAs* e *owl:InverseFunctionalProperty*.

Owl:sameAs è utilizzato per collegare individui separati e affermare che due URI hanno la stessa identità e quindi si riferiscono alla stessa risorsa.

```
@prefix default : <http://www.mml.arces.unibo.it/fvergari/onto#> .
@prefix rdfs : <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl : <http://www.w3.org/2002/07/owl#> .
default : Fabio
  a default : Person .
default : Federico
  a default : Person ;
  = default : Fabio.
```

Owl:InverseFunctionalProperty dichiara che l'oggetto di una proprietà determina univocamente il soggetto. Una proprietà P, che è inversamente funzionale, asserisce che l'oggetto O è l'oggetto di P solo per il soggetto S.

```
@prefix default : <http://www.mml.arces.unibo.it/fvergari/onto#> .
@prefix rdfs : <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl : <http://www.w3.org/2002/07/owl#> .
default : Fabio
  a default : Person .
default : Luciano
  a default : Person ;
  default: biologicalFatetherOf
    default: Fabio .
default : biologicalFatherOf
```

*a owl:ObjectProperty , owl:InverseFunctionalProperty ;
rdfs : domain default:Person ;
rdfs: range default : Person .*

Questo esempio afferma che Luciano è il padre biologico di Fabio. Poiché la proprietà *biologicalFatherOf* è inversamente funzionale, Luciano può essere il solo padre biologico di Fabio.

Una ragione per limitare l'espressività del linguaggio è che i reasoners OWL-DL possono essere molto costosi in termini di complessità e di tempi computazionali. L'RDF++ fornisce comunque la *deductive closure* sul grafo RDF++ richiedendo meno risorse computazionali. Con riferimento a Lassila [61], la *deductive closure* è un requisito base minimo per sistemi basati su RDF e l'RDF++ fornisce questa funzionalità con un' impatto minimo sulle performance.

Un altro requisito per l'interoperabilità dei dati in sistemi basati su un'ontologia è che i sistemi condividano la stessa ontologia e lo stesso modo di utilizzarla. Per facilitare e aiutare comunità nella condivisione di informazioni e nella comprensione del significato semantico dell'informazione diverse ontologie sono state rilasciate.

4.4 Ontologie esistenti

Attualmente diverse ontologie possono essere trovate pubblicamente. Queste possono essere divise inizialmente in due grosse categorie: ontologie generali e ontologie specifiche per domini applicativi.

Le ontologie generali forniscono definizioni per termini general-purpose, e possono essere usate per costruire ontologie per domini specifici. Queste ontologie modellano concetti chiave che sono usati anche in specifici domini fornendo il punto di partenza per creare ontologie specifiche; utilizzando un'ontologia generale come punto di partenza per ontologie specifiche è possibile standardizzare la rappresentazione delle informazioni tra domini applicativi differenti rendendo possibile la condivisione di concetti e raggiungendo un certo livello di interoperabilità.

Due famose ontologie sono SUMO [62] e DOLCE [63]. La Suggested Upper Merged Ontology è candidata a diventare l'ontologia standard dell'IEEE. L'obiettivo è di contenere un vocabolario di 1000 – 2500 termini chiave per permettere di modellare domini specifici. Nel tempo l'ontologia SUMO completa conterrà varie ontologie specifiche per un totale di 70000 termini. L'ontologia DOLCE Ontology for Linguistic and Cognitive Engineering non è invece candidata a essere un'ontologia universale, ma nasce come una fondazione finalizzata alla raccolta delle entità ontologiche tipiche e nascoste nel linguaggio naturale e nel senso comune.

Le ontologie specifiche focalizzano invece i loro concetti e strutture su determinati domini applicativi. I campi di interesse sono molto vasti spaziando dal campo biomedico alle comunicazioni. L'ontologia più citata per quanto riguarda il campo dei sensori è OntoSensor [64]. OntoSensor non fornisce specifici dettagli per sensori appartenenti ad un determinato dominio applicativo ma invece fornisce una modellazione per la descrizione dei sensori e dei dati. OntoSensor è basata su SUMO e estende alcuni concetti per descrivere i sensori e i dati raccolti seguendo al tempo stesso gli standard SensorML [65] e ISO 19115 [66]. L'obiettivo di OntoSensor è dunque quello di creare un linguaggio comune per la definizione, classificazione e descrizione di sensori. Un'altra ontologia nota per la descrizione di reti di sensori può essere trovata in [67], nata originariamente per descrivere dati di una rete di sensori. Anche questa ontologia è un'estensione di SUMO e nello specifico si tratta di tre distinte ontologie: Sensor Data Ontology, la Sensor Hierarchy Ontology e la Extension Plug-ins Ontology. Le prime due descrivono i sensori e i dati, mentre l'ultima permette di estendere l'ontologia per specifici domini applicativi.

5 Spazi Intelligenti e piattaforme di supporto

Gli Smart Spaces sono utilizzati per raccogliere dinamicamente dati locali che portano informazione sul mondo reale (come la temperatura di una stanza, le condizioni fisiologiche di un individuo, il traffico di una strada) e mettono a disposizione tali dati per applicazioni che possono beneficiarne. Uno SS è una parte di uno Smart Environment poiché è il componente centrale usato per condividere e memorizzare l'informazione che necessita di essere distribuita in un ambiente. Diversi progetti di ricerca su Smart Environment sono presenti in letteratura. Lo schema di Cook è mostrato in figura 7.



Figura 7 Schema di Smart Environment (Cook)

Alla base di uno SE è presente un ambiente "connesso" che permette ai dispositivi di comunicare facilmente. Sopra il livello di comunicazione, sensori e attuatori permettono al sistema di comprendere ciò che sta accadendo nell'ambiente per agire di conseguenza. Al centro di tutto si trovano dispositivi intelligenti, apparecchiature e, più importante di tutti, la persona che interagisce e vive nell'ambiente.

Come già citato diverse soluzioni sono già presenti e consentono di costruire ambienti che rilevano l'utente mediante diverse tecnologie (video, audio,..) e

forniscono vari servizi; il problema principale è la mancanza di interoperabilità avendo ogni realizzazione un proprio protocollo e modello di database per la rappresentazione dei dati.

5.1 Smart Space e Smart-M3

Lo Smart Space è un'evoluzione dello stato dell'arte attuale per quanto riguarda l'interoperabilità di informazione e tra sistemi differenti.

Come accennato precedentemente, lo SS è un'entità per la condivisione dell'informazione per mezzo di una SIB. In questa tesi, la piattaforma utilizzata è Smart-M3 [68] [69] [70], sviluppata all'interno del progetto SOFIA [71]. Questa piattaforma è simile al concetto di Semantic Spaces presente in [72]. Smart-M3 è stato sviluppato all'interno del progetto europeo SOFIA per realizzare una piattaforma innovativa capace di soddisfare 15 principi [73]. Tra questi i più rilevanti per questa discussione sono i seguenti: *agnostics*, *notification*, *evolvability*, *extensibility* e *legacy principles*.

L' *agnostic principle* afferma che la piattaforma deve essere agnostica rispetto il caso d'uso, le applicazioni, l'ontologia usata per rappresentare l'informazione, i linguaggi di programmazione, i servizi e i livelli di comunicazione usati per esporre lo SS. Il *notification principle* assicura che un client possa sottoscrivere per essere allertato al verificarsi di determinati eventi. L' *evolvability principle* assicura che la piattaforma provveda la capacità di poter evolvere il sistema sostenendo una riconfigurazione dinamica quando cambiano le informazioni all'interno dello SS. Per esempio se nuovi sensori sono aggiunti allo SS, l'applicazione può beneficiarne senza il bisogno di modificare il codice esistente. L' *extensibility principle* assicura la possibilità di poter aggiungere estensioni sia a livello di informazione che di servizio alla piattaforma; appropriate ontologie e modelli di rappresentazione dei dati possono essere usate per esempio per aggiungere un controllo degli accessi e definire politiche di controllo e privacy; la sicurezza e la privacy che sono qualità importanti in uno scenario di telemedicina possono essere introdotte come estensioni. Il *legacy principle* afferma che i dispositivi esistenti già in commercio (*legacy*) possono scambiare informazione con lo SS attraverso moduli di interfacciamento chiamati *legacy adapters*.

Smart-M3 è costituito da due componenti principali: il Semantic Information Broker e il Knowledge Processor.

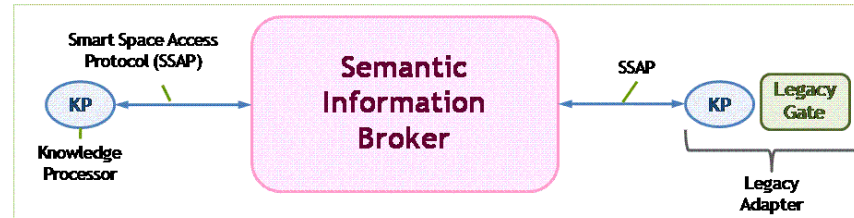


Figura 8 Architettura di Smart-M3

Allo stato attuale uno Smart Space è costituito da una sola SIB ma la visione prevede di distribuire l'informazione su più SIB. La SIB è dunque un'entità digitale che memorizza e condivide le informazioni del mondo reale tenute costantemente in aggiornamento; le informazioni contenute nella SIB rappresentano dunque una fotografia che rappresenta la situazione attuale dell'ambiente. L'informazione è memorizzata in grafi RDF e guidata da un'ontologia. La SIB offre inoltre un linguaggio di query che fornisce un supporto di reasoning alle applicazioni.

I KP sono componenti software che interagiscono con la SIB producendo e/o consumando informazione. I KP scambiano informazione con la SIB attraverso un protocollo XML chiamato SSAP (Smart Space Application Protocol) che definisce un semplice set di messaggi.

- *Join* : è la prima operazione che deve essere eseguita da un KP e consiste nella registrazione all'interno dello SS;
- *Leave* : è l'ultima operazione che viene eseguita dal KP e rappresenta il fatto che il KP non eseguirà più operazioni con la SIB;
- *Insert*: consente al KP di inserire un sottografo;
- *Remove*: consente al KP di rimuovere un sottografo;
- *Update*: è un'operazione che consiste in una operazione atomica di remove e di insert;
- *Query*: consente di prelevare informazioni dalla SIB utilizzando i linguaggi di query supportati;
- *Subscribe*: consente di realizzare una query persistente e in sostanza il KP viene notificato dalla SIB tutte le volte che viene modificata l'informazione alla quale il KP si è sottoscritto;

- *Unsubscribe*: permette di chiudere la sottoscrizione.

I messaggi SSAP possono essere utilizzati su differenti meccanismi di comunicazione come il TCP/IP, Bluetooth o NoTA [32]. Le librerie che implementano le SSAP sono attualmente disponibili in Python, in Java, C#, C++/GlibC, C++/QT and ANSI C. Queste librerie consentono allo sviluppatore di astrarsi dalle interfacce della SIB per lavorare direttamente sulle informazioni a livello di triple su cui operare.

Oltre all'uso di queste librerie i KP possono essere realizzati anche utilizzando l'Application Development Kit (ADK); lo scopo dell'ADK consiste proprio nel facilitare la scrittura di codice per realizzare KP; partendo da una o più ontologie, l'ADK genera del codice per la rappresentazione delle classi definite nell'ontologia in modo tale da permettere allo sviluppatore di programmare con concetti più familiari piuttosto che utilizzare operazioni a basso livello come SSAP e informazioni in formato di triple RDF.

La SIB supporta diversi linguaggi di query tra cui query basata su triple e Wilbur query; è inoltre possibile estendere le funzionalità della SIB aggiungendo altri linguaggi di query.

Le query basate su triple permettono di inserire *wildcard* in una qualsiasi posizione della tripla RDF. Per esempio una query del tipo <any, any, any> richiede tutte le triple disponibili nella SIB. Le risposte alle RDF query sono sempre un elenco di triple.

Il Wilbur Query Language è invece più interessante poiché forniscono come funzionalità la *deductive closure* e permettono di realizzare query più complesse. Infatti il WQL supporta cinque tipi di query.

- *Values*: consiste in un nodo di partenza e in un'espressione per il percorso; la risposta alla query è un elenco che contiene tutti i nodi che sono raggiunti attraverso il percorso specificato a partire dal nodo di partenza;
- *Related*: consiste in un nodo di partenza, un'espressione per il percorso e in un nodo finale. La risposta è un booleano che assume valore *True* se esiste il sottografo cercato;
- *Nodetypes*: consiste in un nodo. La risposta della query è la gerarchia delle classi a cui appartiene il nodo;

- *Istype*: consiste in un nodo di partenza e un nodo per specificare un tipo. La risposta è un booleano che assume valore *True* se il nodo appartiene alla classe specificata nella query.
- *Issubtype*: consiste in un nodo per il sottotipo e un per il supertipo. La risposta è un booleano che assume valore *True* se è verificata la condizione richiesta.

I percorsi nelle espressioni delle Wilbur query consentono di navigare all'interno del grafo. La SIB permette inoltre allo sviluppatore di sottoscrivere a diverse parti del grafo e ricevere una notifica ogni volta che si verifica una modifica in tale sottografo. Questo grazie alle sottoscrizioni che permettono di realizzare delle query persistenti; infatti quando l'informazione di interesse è modificata il KP è notificato e nel messaggio di notifica è presente la parte di grafo rimossa e quella aggiunta.

6 Disegno dell' ontologia

L'interoperabilità è considerata un requisito per realizzare innovativi sistemi informativi in qualsiasi dominio. Nell'ultimo decennio la ricerca in questo ambito ha mostrato come le ontologie siano un caposaldo per raggiungere interoperabilità a livello di informazioni. In questo capitolo sarà descritto lo sviluppo dell'ontologia realizzata e usata nello Space Space descritto nel prossimo capitolo. Infatti l'utilizzo di un'ontologia è un punto fondamentale per garantire che l'informazione contenuta in uno Smart Space sia interoperabile. Lo sviluppo dell'ontologia è un lavoro parallelo allo sviluppo dello Smart Space.

6.1 Metodologie per lo sviluppo di una ontologia.

Il progetto di un'ontologia per uno Smart Space è un passo fondamentale per la comune comprensione della gerarchia dell'informazione tra vari agenti software e programmatori.

Diverse metodologie per il design di ontologie sono presenti in letteratura [74], [75], [76], [77]. Un semplice approccio non focalizzato ad un determinato dominio ma generico è quello proposto da Noy [77]; questo approccio propone una serie di passaggi guida e sarà utilizzato nella discussione.

Finalità dell'ontologia.

Il punto di partenza consiste nell'analizzare per cosa sarà utilizzata l'ontologia. Dato un determinato dominio, bisogna analizzare il tipo di conoscenza che si vuole rappresentare. Nel caso di uno Smart Space è importante ricordare chi e cosa sono gli attori principali del sistema e quindi considerare quello che il sistema deve fare.

Per esempio in uno scenario di tele-monitoraggio, gli attori sono i pazienti, i medici, i sensori e i dati e le relazioni tra essi. In base alle funzionalità che si vogliono raggiungere è possibile includere ulteriori concetti come per esempio quello degli allarmi.

Domande di competenza.

Partire dallo scopo dell'ontologia permette di capire a quali domande il set di informazioni deve poter fornire una risposta all'interno dello scenario. Utilizzando questo insieme di domande è possibile avere dei requisiti di partenza che l'ontologia dovrà soddisfare per essere completa; inoltre il set di domande facilita lo sviluppatore dell'ontologia nella creazione dei tipi di classi e delle loro relazioni.

Analisi delle ontologie esistenti.

L'utilizzo di un'ontologia già esistente è sempre un valore aggiunto. Infatti se un team di esperti ha già realizzato un'ontologia utile per la descrizione dello stesso dominio, l'utilizzo della stessa ontologia aumenta l'interoperabilità del sistema. Il linguaggio specifico con cui è creata l'ontologia non rappresenta necessariamente un problema in quanto gli editor di ontologie come Protege supportano differenti formati.

Elenco dei termini importanti.

Questo step consiste nel fare un elenco di tutti i termini e relazioni (frasi) che si vogliono rappresentare. Per ogni termine è utile definire le proprietà e cosa serve per descriverlo.

Scelta del linguaggio e del tool per definire l'ontologia.

Una scelta prudente è sempre quella di partire dal fatto di considerare che l'ontologia sarà utilizzata in applicazioni future. Per mantenere l'interoperabilità è quindi consigliato l'utilizzo del linguaggio OWL. Scelto il linguaggio è importante anche il tool utilizzato per editarla. Nonostante sia possibile scrivere a mano un'ontologia, il tool semplifica la creazione dell'ontologia fornendo utili funzionalità come la visualizzazione e il supporto al reasoning.

Definizione delle classi e delle gerarchie tra le classi.

L'approccio per la definizione delle classi è simile a quello utilizzato nel progetto di sistemi. Può essere utilizzato un approccio di tipo top-down partendo dai concetti più generali per poi scendere negli elementi più specifici. Viceversa un approccio di tipo bottom-up parte dalle classi più specifiche e richiede di raggrupparle in concetti più generici. Una combinazione dei due

approcci è una valida procedura definendo passo dopo passo concetti specifici e generali. Questo step si basa pesantemente sull'elenco dei termini da rappresentare e ha l'obiettivo di raggrupparli e categorizzarli.

Definizione delle proprietà.

Definite le classi, devono essere create le strutture per rappresentare i concetti. Infatti le gerarchie tra classi non rispondono a tutte le domande precedentemente raccolte. Per poter soddisfare questo requisito è necessario introdurre proprietà tra classi. Nel caso dell'OWL le proprietà collegano fra loro individui ma non direttamente le classi.

Definizione di dominio e range delle proprietà.

Una proprietà è una relazione tra due risorse. La relazione può avere delle restrizioni definendo un dominio ed un range. Questo limita l'utilizzo di una determinata proprietà solo a certe classi impedendo un errato utilizzo. Il concetto di sottoclasse nella gerarchia delle classi è applicato anche al dominio ed al range. Per esempio se la proprietà *identifiedby* ha dominio *objects* e range *id*, allora tutti le sottoclassi di *objects* possono utilizzare la proprietà *identifiedby*. Il dominio ed il range di una proprietà sono fondamentali per aiutare i non esperti nell'utilizzare correttamente l'ontologia garantendo coerenza semantica.

Creazione delle istanze.

La creazione delle istanze, o individui, è un passo fondamentale per capire se con l'ontologia realizzata è possibile rappresentare tutti i concetti che si vogliono descrivere. Questo step necessita delle proprietà e delle classi definite nei passi precedenti. Qui il designer deve tentare di mappare i concetti a livello più basso ovvero deve riempire l'ontologia con individui e relazioni usando i concetti precedentemente introdotti. Questo step è utile anche per creare una documentazione dell'ontologia in modo tale da spiegare come combinare opportunamente i concetti forniti dall'ontologia.

Valutazione dell'ontologia.

Questo è un passo non banale. A questo punto infatti entrano di nuovo in gioco le domande di competenza e la lista dei termini rappresentati; l'obiettivo di questo step è valutare se è possibile rispondere alle domande utilizzando l'informazione disponibile nell'ontologia.

Iterazione.

Per sviluppare un'ontologia che si adatti il più possibile alla descrizione di un dominio è infatti richiesto un approccio iterativo. Data una bozza iniziale dell'ontologia e valutata, è importante ripetere i passi precedenti e correggere i problemi riscontrati.

6.2 Progetto dell'ontologia per Smart Space

Per realizzare l'ontologia del nostro Smart Space abbiamo utilizzato la metodologia sopra illustrata. La nostra ontologia non è finalizzata a rappresentare tutti i concetti di un determinato dominio; l'idea è invece creare un'ontologia base per lo sviluppo di un prototipo di Smart Space. Durante lo sviluppo dell'ontologia, è stato utilizzato un approccio iterativo: partendo da una analisi di tipo top-down abbiamo determinato i concetti astratti da rappresentare; quindi abbiamo iterato con una visione bottom-up del sistema per capire quali informazioni dovevano essere fornite.

Scopo. Lo scopo della nostra ontologia è quello di descrivere lo stato corrente dei dati sensoriali e le informazioni generiche che caratterizzano una applicazione Smart Space. Essendo un'ontologia da utilizzare per lo sviluppo di un prototipo di Smart Space, un obiettivo fondamentale è stato quello di renderla facilmente estendibile in modo tale da poter integrare nuovi concetti senza modificare l'ontologia già esistente prevedendo quindi di aggiungere in sviluppi futuri nuovi tipi di sensori e funzionalità. Come punto di partenza, i concetti da rappresentare sono quelli che permettono la descrizione di dati ambientali (inclusi la temperatura, l'umidità), la localizzazione (sia di utenti che di dispositivi), parametri fisiologici e informazioni base sul profilo di utenti e dispositivi. Ogni dato nell'ontologia deve essere descritto da alcuni parametri tra cui il tipo di misurando (cioè il tipo di misura, come *HeartRate*, *PostureAngle*, *RelativeHumidity*), unità di misura e relazioni che lo legano all'entità che lo ha prodotto. Per quanto riguarda la localizzazione nel sistema è limitata alla granularità di una stanza.

Domande di competenza. Queste domande saranno utili per testare l'ontologia alla fine del suo ciclo di sviluppo.

Quale è l' HR di Person_01?

Dove è Person_01?

Chi è presente nella Room_01?

Qual è la temperatura nella Room_01?

Quali sono i dispositivi nella Room_01?

Quali sono i dispositivi di Person_01?

Qual è il profilo di Person_01?

Quali sono temperatura e umidità nell'ambiente in cui si trova Person_01?

Quale è il MACAddress di SensorPlatform_01?

Quale è il protocollo del dispositivo con MACAddress_01?

Da una prima analisi di queste domande è facile capire come sia necessaria una descrizione dei dispositivi e dei dati sensoriali ed una loro associazione all'utente e all'ambiente.

Analisi delle ontologie esistenti. Come accennato la nostra ontologia non copre tutti i concetti di un determinato dominio essendo finalizzata a coprire i concetti per descrivere applicazioni Smart Space. Un punto di partenza è stato comunque OntoSensor usato come punto di partenza per definire il concetto di dato.

Lista dei termini importanti. L'elenco dei termini principali da rappresentare è basata sulle finalità della nostra applicazione.

Data

Identification Data

Sensor Data

Safety Limit

Event Type

Device

Sensor Platform

Environment

Event

Measurand
Person
Unit of Measure

Scelta del linguaggio e del tool per definire l'ontologia. Il linguaggio scelto per la nostra ontologia è l' RDF++ serializzato in standard XML/RDF. Questa scelta è stata motivata dalla piattaforma utilizzata ovvero da Smart-M3 che offre reasoning su RDF++. Il tool utilizzato per lo sviluppo e l'editing dell'ontologia è stato Protegè; il tool è risultato conveniente per effettuare modifiche all'ontologia e per i plug-in che mette a disposizione che permettono di avere un visualizzazione grafica dell'ontologia.

Definizione delle classi e della gerarchia fra classi. L'albero delle classi è mostrato in figura 9.

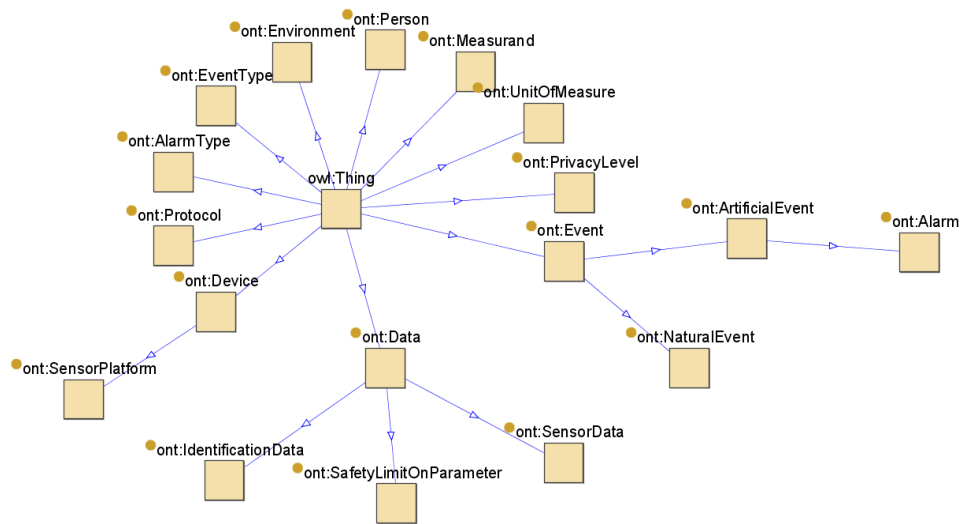


Figura 9 Albero delle classi

La classe *Data* è utilizzata per descrivere un generico dato prodotto da un dispositivo o sensore. Un' istanza di *Data* può essere associato ad un utente o ad un ambiente e può essere anche prodotto in seguito ad un'elaborazione; un esempio può essere la temperatura media di un ambiente che è monitorato da più sensori di temperatura. *SensorData* è una sottoclasse di *Data* utilizzata per rappresentare il dato grezzo ottenuto da un sensore. Per esempio una lettura del sensore di temperatura è un *SensorData* ed è descritto da un valore, da un timestamp ed è collegato a istanze di *MeasurandType* e *UnitOfMesasure*. *UnitOfMeasure* contiene le istanze delle varie unità di misura usate nel siste-

50

ma. La classe *Measurand* descrive i diversi tipi di misurandi. Sviluppando in questo modo il concetto di *Data* si ottiene una soluzione molto flessibile ed estendibile: quando è aggiunto un nuovo sensore che misura una grandezza non ancora descritta basta aggiungere nuove istanze per caratterizzarla.

La classe *IdentificationData* è usata per descrivere l'identificatore necessario per identificare una persona, un dispositivo o un generico oggetto; una istanza di *IdentificationData* può avere un valore e un tipo (per esempio RFID o MACAddress o SemaCode). La classe *SafetyLimitOnParameter* è usata per definire delle soglie come limiti di sicurezza per determinati tipi di dati. Per esempio è possibile settare dei valori di massimo e minimo per determinati parametri come la temperatura in un ambiente o il peso per un paziente. La classe *Device* è usata per descrivere un dispositivo generico come un PC o uno smartphone; la sua sottoclasse *SensorPlatform* è stata invece introdotta per descrivere una piattaforma di sensori come per esempio l'Mote2.

La classe *ProtocolType* descrive i protocolli utilizzati da una *SensorPlatform*.

Environment descrive un ambiente. La classe *Event* ha due sottoclassi *ArtificialEvent* e *NaturalEvent*; è usata per descrivere un evento come può essere per esempio un allarme creato quando un parametro eccede la soglia di sicurezza. *EventType* specifica il tipo di evento.

La classe *Person* descrive gli utenti (pazienti e medici). La classe *PrivacyLevel* è stata introdotta per settare livelli di privacy a parti dell'informazione.

Definizione delle proprietà.

Le proprietà sono mostrate in tabella.

Proprietà	Dominio	Range
ContainsEntity	Environment	
HasData		Data
HasEventType	Event	EventType
HasIdentificationData		IdentificationData
HasIdentificationType	IdentificationData	IdentificationType
HasMeasurand	Data	Measurand
HasPrivacyLevel		PrivacyLevel
HasSafetyLimitOnParameter		SafetyLimitOnParameter
HasSensorData	Device	SensorData

HasUnitOfMeasure	Data	UnitOfMeasure
IsRelatedToPerson		Person
IsRelatedToSensor		SensorPlatform
IsWearing	Person	
HasProtocol	Device	ProtocolType
HasBTPin	Device	StringLiteral
HasExpirationTime	Data	StringLiteral
HasFriendlyName		StringLiteral
HasMACAddress	Device	StringLiteral
HasName	Person	StringLiteral
HasText		StringLiteral
HasTimeStamp	Data	StringLiteral
HasValue	Data	StringLiteral

La *ContainsEntity* è usata per collegare un *Environment* con una *Person* o un *Device*. Questo link viene creato per definire la posizione di una persona o di un dispositivo.

HasData è usato per collegare una qualsiasi istanza che può avere un *Data* con il rispettivo *Data*.

HasMeasurand collega un *Data* ad un *Measurand*; è utile per esempio per effettuare query quando si è interessati solo ai dati con un determinato significato fisico.

HasSensorData è usato in modo simile a *HasData* ma collega tra loro *SensorPlatform* a *SensorData*.

HasUnitOfMeasure connette *Data* a *UnitOfMeasure*.

HasEventType è usato per collegare una istanza di *Event* al rispettivo tipo.

HasIdentificationData collega una generica istanza ad un *IdentificationData*.

HasIdentificationType connette un *IdentificationData* con *IdentificationType*.

HasSafetyLimitOnParameter collega o un *Environment* o una *Person* a una istanza di *SafetyLimitOnParameter*.

IsWearing è usato per collegare una persona con un dispositivo o un sensore.

IsRelatedToPerson e *IsRelatedToSensor* è usato per collegare rispettivamente a una *Person* o a un *Sensor* una qualsiasi istanza.

HasProtocol collega un *Device* ad un *ProtocolType*.

Le seguenti sono invece tutte le datatype property.

HasBTPin associa il pin bluetooth al dispositivo.

HasExpirationTime è usato per dare un tempo di validità ad un generico dato.

HasFriendlyName associa una stringa ad una istanza in modo da facilitare la sua ricerca.

HasName è usata per linkare una *Person* al suo nome.

HasText è usato per aggiungere una descrizione testuale a una qualsiasi entità.

HasTimestamp associa un timestamp ad un *Data*

HasValue associa un valore ad un *Data*.

Definizione del dominio e del range delle proprietà.

Il dominio ed il range delle proprietà sono mostrati in tabella e sono spiegati per ogni proprietà nella sezione precedente.

Creazione delle istanze.

In questo passo sono state create tutte le istanze. Per ogni istanza è stata utilizzata la seguente notazione:

ClassName_UUID dove ClassName è il nome della classe a cui appartiene l'istanza mentre UUID è un identificativo random (universally unique identifier). Questo garantisce l'unicità di ogni URI nel sistema.

Valutazione dell'ontologia.

Guardando lo scopo dell'ontologia e le classi, le proprietà e le istanze che sono state create, l'ontologia permette di rappresentare tutte le informazioni rilevanti nella nostra applicazione ed risulta essere una buona base per la realizzazione del nostro prototipo di SS. L'ontologia permette di rispondere alle domande di competenza e di utilizzare i termini importanti.

Iterazione.

Questo processo richiede che l'ontologia sia usata nell'implementazione che sarà descritta nel capitolo successivo.

6.3 Conclusioni

Lo sviluppo di un'ontologia è un procedimento qualitativo e non esiste una procedura unica per la sua realizzazione; non esiste dunque una soluzione unica e differenti soluzioni possono essere altrettanto valide. La metodologia di [77] aiuta lo sviluppatore suggerendo una linea guida tramite dei passi da seguire durante questo processo.

7 Progetto del prototipo di Smart Space per il monitoraggio concorrente di parametri fisiologici ed ambientali

Lo scopo di un sistema Smart Space è quello di permettere l'integrazione di sistemi diversi e dati eterogenei in modo tale da favorire collaborazione e sviluppo di applicazioni innovative. La funzionalità dello SS di riflettere l'informazione del mondo reale in uno spazio digitale e la condivisione di queste informazioni con differenti sistemi sono un passo innovativo nello sviluppo di sistemi interoperabili. Sono molti gli ambiti e gli scenari in cui i concetti degli SS possono essere applicati, variando da una casa intelligente, a un monitoraggio medico, a tutte le situazioni in cui una condivisione di informazioni porta a dei benefici.

Il prototipo di SS descritto in questo capitolo è stato progettato per monitorare i parametri fisiologici di un utente e, in modo concorrente, le condizioni ambientali in cui esso si trova.

7.1 Progettazione di uno Smart Space

La progettazione di un sistema SS è molto simile ai passi seguiti per il design di un sistema standard. Durante i vari steps è stato necessario effettuare particolari considerazioni qui elencate.

1. *Requisiti.* Come per ogni sistema, il design incomincia capendo quello che il sistema deve fare. Inoltre in un sistema SS è fondamentale considerare quale è il ruolo della condivisione dell'informazione. Ci si deve chiedere tre domande.
 - a. Il sistema deve cooperare con sistemi già esistenti? In questa situazione è necessario considerare se è possibile realizzare degli adattatori per rendere i dati forniti dal sistema già esistente disponibili nello SS.

- b. Il sistema deve cooperare con altri SS? Se si è necessario capire se è riutilizzabile la stessa ontologia o se risulta più conveniente crearne una nuova. Se è necessario crearne una nuova e si vuole beneficiare dell'informazione dei due sistemi, è necessario mappare i concetti della precedente ontologia in quella nuova riuscendo a replicare in modo differente le stesse informazioni.
 - c. Che tipo di informazione è condivisa ed è una buona soluzione per i sistemi che utilizzeranno i dati accedere allo SS? Con requisiti di efficienza si può optare per soluzioni più performanti perdendo interoperabilità (per esempio usando database); trattando dati sensibili è inoltre necessario garantire sicurezza e privacy.
2. *Specifiche.* Oltre alle specifiche standard sul livello software e hardware, è qui che il designer inizia a considerare lo sviluppo dell'ontologia. Infatti l'ontologia è una parte fondamentale di uno SS ed è in base ad essa che viene determinato come disaccoppiare il sistema in diversi knowledge processor. Le specifiche dovrebbero inoltre suggerire il middleware da utilizzare così come i diversi standard di comunicazione da supportare. Inoltre se informazioni private saranno condivise nello SS devono essere inclusi metodi o estensioni atti a garantire protezione.
3. *Architettura.* Il design dell'architettura richiede di disaccoppiare i diversi knowledge processor in base alle funzionalità considerando anche possibili sviluppi del sistema. Per esempio considerando l'uso di un sensore, l'elaborazione del dato grezzo piuttosto che essere integrata localmente direttamente nel KP del sensore, può essere effettuata da un KP separato; quest'ultima soluzione favorisce il riutilizzo del dato grezzo per futuri algoritmi lasciando intatto il sistema.
4. *Implementazione.* L'implementazione dei diversi KP e dell'hardware necessario per realizzare il sistema SS è il task finale.

Una volta ottenuta la visione globale del sistema con i suoi requisiti, le specifiche e l'architettura, ogni KP può essere realizzato.

Queste considerazioni sono le principali differenze tra lo sviluppo di un sistema standard ed un sistema SS; la linea guida generale è di rendere il sistema il più interoperabile, estensibile e aperto possibile. Condividendo tutti i dati prodotti dal sistema o da più sistemi in uno SS si crea il potenziale per creare applicazioni innovative.

7.2 Prototipo di Smart Space

In questo capitolo sarà discussa la realizzazione del nostro sistema SS che integra informazioni ottenute da differenti sensori commerciali e permette agli agenti software di beneficiare dell'interoperabilità offerta dal sistema.

7.2.1 Motivazioni del prototipo

Lo scenario proposto si basa su concetti presenti in CHIRON [78]. Con questo progetto è condiviso lo scenario e l'obiettivo di realizzare un sistema per il controllo della salute personalizzato; la realizzazione di questo obiettivo riguarda chiaramente più domini coinvolgendo più attori tra cui il paziente, i medici e la comunità scientifica. L'obiettivo di CHIRON è di progettare una architettura per realizzare effettivamente un sistema per il person-centric health management in grado di seguire il paziente in tutte le sue cure. La principale sfida è quindi integrare le informazioni più recenti del paziente con gli esami e i dati storici e trasformare questa informazione in un utile supporto per facilitare la presa di decisioni. Per realizzare questa visione diversi requisiti devono essere soddisfatti. Prima di tutto i dati devono essere raccolti da fonti eterogenee (sensori, archivi, database). Inoltre per essere resi comprensibili e disponibili in modo semplice devono essere rappresentati in modo uniforme in una piattaforma per la condivisione dell'informazione. Memorizzati i dati, devono essere realizzati meccanismi per la lettura e l'analisi. In tutto questo processo è necessario una rappresentazione dell'informazione per armonizzare il processo che va dal controllo del paziente al supporto offerto allo staff medico nella presa

delle decisioni. Nella architettura di CHIRON vengono definiti i tre seguenti layer: l'*user plane*, il *medical plane*, lo *statistical plane*.

- Lo *user plane* riguarda le interazioni da e con il paziente (monitoraggio e feedback locale).
- Il *medical plane* riguarda le interazioni da e con i medici (valutazione dei dati clinici, diagnosi, pianificazione e esecuzione di un trattamento, feedback al paziente)
- Lo *statistical plane* riguarda le interazioni con i ricercatori medici (controllo e raccolta dati).

L'intero ciclo di cura è basato sulla continua interazione e scambio di informazioni tra questi tre piani.

L'utilizzo di Smart-M3 e di uno SS system permettono di dare un evidente contributo alla creazione di un *integrated care cycle* fornendo il controllo di uno spazio di informazioni contenente dati ottenuti da sensori portabili e stazionari, supportando strumenti di misura e attuatori diversi ed eterogenei, abilitando un profilo personale ed un approccio personalizzato nell'ambito dell'healthcare.

7.2.2 Requisiti del prototipo

Il nostro scenario consiste nel considerare alcuni utenti che vivono e si muovono liberamente in un ambiente monitorato da sensori ambientali. L'utente ha a disposizione alcuni dispositivi personali e tra questi anche dispositivi medici, alcuni di essi indossabili. L'utente si muove liberamente in questo smart environment e i suoi dati di salute insieme ai dati del suo contesto sono continuamente monitorati.

Il primo requisito per realizzare uno SS di questo tipo è creare una descrizione di tutte le entità che giocheranno un ruolo significativo nel sistema. Per mappare ogni entità fisica nella rappresentazione digitale, ogni entità fisica deve essere identificata univocamente. Quindi ogni entità deve essere descritta con l'appropriato modello in base l'ontologia. Questa descrizione è creata durante l'inizializzazione del sistema e riguarda gli utenti (medici e pazienti), i loro profili, i dispositivi, gli ambienti e tutte le relazioni tra essi. Le relazioni tra queste entità possono essere aggiunte o cambiare dinamicamente; questa

funzionalità può essere realizzata manualmente o in modo automatico con sistemi di identificazione o localizzazione. Per esempio nello scenario proposto, l'associazione dello stesso set di strumenti a diversi pazienti può essere effettuato manualmente tramite un software con GUI o in modo automatico utilizzando la tecnologia RFID. I dispositivi medici e un RFID reader sono posizionati su un banco di misure; il paziente si registra semplicemente posizionando il suo RFID tag sull'RFID reader; quindi il sistema in automatico associa tutte le misure acquisite dai dispositivi di quel banco a quel paziente fino a quando l'RFID viene rimosso.

La architettura è basata su Smart-M3 e è descritta in figura 10.

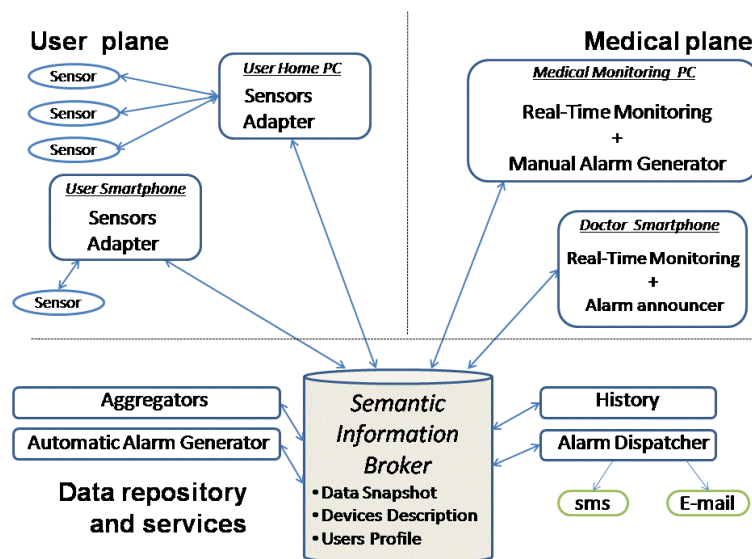


Figura 10 Visione generale del sistema SS

Nello User plane, dati provenienti da sensori eterogenei sono raccolti da un PC o uno smartphone e inviati alla SIB. I *Sensors Adapters* mostrati in figura 10 sono *legacy adapters*; possono essere visti come traduttori guidati da un'ontologia consentendo lo scambio di informazione tra lo SS il mondo legacy; per mondo legacy intendiamo dispositivi e sistemi commerciali già esistenti e presenti sul mercato. La SIB è il nucleo del sistema; memorizza e condivide non solo i dati prodotti dai sensori ma tutte le informazioni create durante il processo di inizializzazione. Questa implementazione è coerente con il concetto di *Alter Ego* proposto in CHIRON. *Alter Ego* è un'entità virtuale dinamica che modella tutte le informazioni rilevanti legate allo stato di salute di un paziente includendo la sua storia, le abitudini, i suoi parametri fisiologici,.. Questa entità deve avere la capacità di evolversi adattandosi a vari domini e

condizioni di utilizzo. Una volta generato questo profilo virtuale è costantemente aggiornato attraverso le informazioni del sistema di monitoraggio; in ogni caso il medico può aggiungere informazioni in qualsiasi momento. Il profilo include informazioni statiche, semi-statiche e dinamiche e su di esso potranno lavorare motori a regole per determinare la presenza di situazioni anomale.

Utilizzando i dati sensoriali collezionati e sfruttando la conoscenza delle relazioni tra le entità interessate, è possibile creare diversi tipi di servizi. Con riferimento alla figura saranno mostrati alcuni esempi.

Gli aggregatori sono servizi generici che consumano l'informazione già presente nella SIB arricchendo questa informazione tramite regole di inferenza. La nuova informazione è quindi memorizzata nella SIB e condivisa in modo tale da poter essere riutilizzata da altri servizi e include per esempio indici e parametri.

L'*Automatic Alarm Generator* è un servizio per la generazione automatica di uno stato di allarme nel profilo dell'utente; questo è generato quando un parametro del paziente esce dal range di sicurezza descritto nel suo profilo.

Il servizio di *History* colleziona i dati rilevanti in modo da consentire una analisi successiva per scoprire o validare relazioni macroscopiche tra il profilo del paziente, la diagnosi e l'efficacia del trattamento medico.

L'*Alarm Dispatcher* è un servizio che notifica la presenza di uno stato di allarme al medico attraverso dei mezzi di comunicazione standard cioè tramite SMS o e-mail.

In riferimento al medical plane abbiamo considerato la seguente politica. Lo staff medico può visualizzare tutti i dati dei suoi pazienti attraverso diverse piattaforme (smartphone e PC) in tempo reale. I medici possono visualizzare questi dati e gli specialisti sono notificati in caso di allarme. I medici possono visualizzare e modificare il profilo del paziente e possono creare degli stati di allarme manualmente.

Il sistema necessita di condividere informazione eterogenee da diversi sistemi/dispositivi commerciali. Lo scopo è creare un sistema che permetta allo staff medico di monitorare lo stato di salute di un paziente insieme ai parametri ambientali. Il sistema è finalizzato al monitoraggio fuori dalla struttura ospedaliera per persone soggette a malattie cardiovascolari. Il sistema dovrà monitorare parametri fisiologici; tra questi la frequenza cardiaca, la frequenza respiratoria, la temperatura della pelle, la pressione arteriosa, il peso, la satura-

zione dell'ossigeno e l'attività del paziente. Questi dati prodotti da diversi dispositivi saranno raccolti da uno smartphone associato all'utente o da un PC. Il sistema di monitoraggio dovrà essere il meno invasivo possibile e soddisfare criteri di usabilità; l'interazione con i dispositivi dovrà essere il più semplice possibile. Il sistema fornirà inoltre informazioni sui dati ambientali; in particolare sono monitorati temperatura, umidità, luce e la presenza di acqua sul pavimento. I sensori ambientali sono sparsi e fissati negli ambienti in cui l'utente si può muovere. L'informazione di localizzazione è fondamentale in questo sistema e la granularità è limitata al livello di stanza. La localizzazione può essere fornita da più sistemi di localizzazione in modo tale da aumentare la robustezza del sistema. Allo stesso modo diversi sistemi possono monitorare la situazione ambientale per aumentare la ridondanza di dati. I dati ambientali dovranno essere utilizzati per astrarre un indice di disagio bioclimatico da associare ad ogni ambiente. Degli allarmi basati su soglie saranno considerati per avvertire il personale medico della presenza di situazioni anomale. Il prototipo non tiene conto di aspetti riguardanti sicurezza e privacy.

7.2.3 Specifiche del prototipo

La soluzione utilizzerà Smart-M3 come supporto per realizzare uno SS. I dati saranno raccolti da diversi sensori. La nostra applicazione si applica ad uno scenario in cui alcuni utenti che muovono in un ambiente indoor diviso in due stanze. Ogni stanza è monitorata da sensori di una Wireless Sensor Network; in particolare è stata utilizzata una WSN con protocollo di comunicazione ZigBee e sviluppata da Eurotech [79]; la WSN è in grado di monitorare temperatura, umidità e l'eventuale presenza di acqua sul pavimento. Per dimostrare l'interoperabilità tra diversi produttori abbiamo aggiunto al sistema anche nodi iMote2 per avere ulteriori misure delle condizioni ambientali.

Per monitorare i parametri fisiologici dell'utente si è scelto di utilizzare un set di dispositivi medici. Questi dispositivi sono gli stessi scelti in collaborazione con il personale medico all'interno del progetto OLDES per il monitoraggio remoto di pazienti con problemi cardiovascolari.

I principali dati fisiologici sono acquisiti dallo Zephyr BioHarness; questo è dispositivo che ha sostituito il nostro prototipo indossabile. Lo Zephyr BioHarness è un dispositivo indossabile; consiste in una fascia con elettrodi tessili ed un dispositivo da abbottonare alla fascia; lo Zephyr BioHarness comunica

tramite Bluetooth, misura diverse grandezze e ha diverse modalità di funzionamento impostabili usando le API fornite dal produttore. Tramite questo dispositivi monitoriamo in tempo reale frequenza respiratoria, frequenza cardiaca, temperatura della pelle e due parametri per misurare l'attività: un indice di attività e l'angolo di postura del paziente.

Oltre a questo dispositivo indossabile, il paziente è dotato anche dei seguenti dispositivi:

- una bilancia pesapersone (A&D UC-321PBT);
- un misuratore di pressione arteriosa (A&D UA-767PBT);
- un pulsiossimetro (Nonin Onyx II9560).

Tutti questi dispositivi comunicano tramite Bluetooth attraverso protocolli per lo scambio di messaggi proprietari.

Il sistema di localizzazione è realizzato sfruttando la tecnologia RFID; è stato utilizzato un TagSense UHF RFID reader per ogni ambiente e un tag è associato ad ogni utente.

I dati forniti dai sensori sono mostrati in tabella. L'ontologia utilizzata nello SS è quella descritta nel capitolo precedente.

Sistema	Measurand	UnitàOfMeasure	Placement
BioHarness	Heart Rate	BPM	Body
BioHarness	SkinTemperature	°C	Body
BioHarness	Posture	°Angle	Body
BioHarness	Activity	VMU	Body
BioHarness	Resp.Rate	BPM	Body
A&D Scale	Weight	Kg	Environment
A&D sphygmomanometer	Systolic Pressure	mmHg	Environment
A&D sphygmomanometer	Diastolic Pressure	mmHg	Environment
A&D sphygmomanometer	Mean Pressure	mmHg	Environment
Nonin pulseoximeter	Oxygen Saturation	%	Environment
Eurotech WSN	Temperature	°C	Environment
Eurotech WSN	Humidity	%	Environment

Eurotech WSN	WaterOnFloor	Boolean	Environment
iMote2	Temperature	°C	Environment
iMote2	Humidity	%	Environment

Per tutti i dispositivi commerciali è stato necessario realizzare dei *legacy adapter*. La funzione di questi adattatori consiste nell'interfacciarsi con il dispositivo applicando il corretto protocollo per lo scambio dei dati, effettuare un parsing dei dati, descriverli con l'ontologia e quindi pubblicarli nello SS.

In questo SS solo interoperabilità a livello di informazione è stata utilizzata. La comunicazione tra i KPs e la SIB avviene utilizzando il protocollo TCP/IP.

Il dispositivo in cui girano i *legacy adapters* per i dispositivi biomedicali deve supportare anche la comunicazione Bluetooth per comunicare con essi. La mobilità dell'utente è garantita dal fatto che il suo dispositivo portatile sia dotato di connettività WiFi.

7.2.4 Architettura del prototipo

L'architettura dello SS è mostrata in figura 11.

La architettura mostra come sia fondamentale il disaccoppiamento di produttori e consumatori di informazione. Inoltre è chiaro come una volta realizzata questa divisione tra KP, ogni KP può essere sviluppato in parallelo allo sviluppo di un'altra parte del sistema.

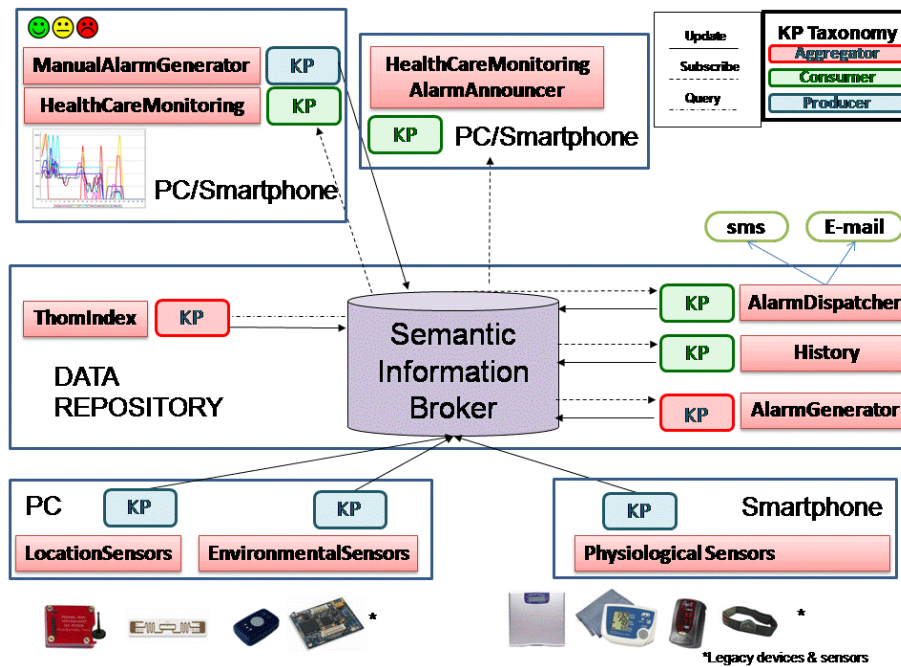


Figura 11 Architettura dello SS

Initialization system KP.

Questo KP non è presente nella figura. Lo scopo di questo KP è quello di inserire tutte le informazioni iniziali nello SS. In particolare registra tutte le entità che hanno un ruolo rilevante e le relazioni tra esse.

Environmental Sensors iMote2 KP

Questo KP è un *legacy adapter* che pubblica nello SS i dati ambientali misurati dai sensori degli iMote2. Ogni mote ha un KP di questo tipo ed ogni mote in fase di inizializzazione è associato ad un ambiente; i dati monitorati cioè la temperatura, l'umidità e la quantità di luce rilevata vengono pubblicate e aggiornate nella SIB e tali misure sono associate all'ambiente in cui è presente il mote. Questo KP gira su un PC con sistema operativo Ubuntu ed è stato realizzato in Python.

Environmental Sensors Eurotech WSN KP

Il *WSN Eurotech KP* è un *legacy adapter* responsabile di pubblicare i dati della WSN Eurotech. Questo KP realizzato in java gira su un PC e coordina e controlla l'intera rete di sensori fungendo da coordinatore. Durante la fase di

inizializzazione ogni sensore è associato ad un ambiente ed il KP aggiorna opportunamente i dati sensoriali misurati; questi sono temperatura, umidità e la presenza di acqua sul pavimento.

Physiological Sensors KP

Il *Physiological Sensor KP* è un *legacy adapter* che si occupa di pubblicare nello SS le informazioni fisiologiche. Questo KP si occupa della comunicazione con tutti i dispositivi biomedicali.

Il KP è l'evoluzione intelligente di distinti KP, uno per strumento. Per soddisfare il requisito di usabilità e semplificare l'interazione dell'utente con il sistema, l'obiettivo è minimizzare il numero di operazioni che l'utente deve effettuare con il sistema. Il modo più semplice e il con minor numero di iterazioni da parte dell'utente per effettuare delle misure, consiste nella seguente procedura: l'utente sceglie uno strumento di misura, lo accende ed effettua la misura; in automatico il sistema associa le misure dello strumento al paziente.

Questa funzionalità è stata raggiunta sfruttando tutta la conoscenza condivisa nello SS. Le informazioni necessarie per realizzare questa funzionalità sono rappresentate dalla descrizione del dispositivo (indirizzo MAC, PIN, protocollo) e dalla relazione semantica che lega il dispositivo all'utente.

Ogni utente infatti ha il proprio *Physiological Sensor KP* che gira sullo smartphone personale dotato di Bluetooth e WiFi o comunque connesso a internet (tramite UMTS). Il KP è stato scritto in Python e gira sia su PC con sistema operativo Ubuntu sia su N900 dotato di sistema operativo Maemo. Il KP comunica con tutti i dispositivi Bluetooth presenti nelle vicinanze e associati all'utente. Infatti il KP esegue le seguenti operazioni: quando lanciato il KP effettua delle query per memorizzare in una tabella interna tutte le informazioni sui dispositivi associati all'utente; tra queste informazioni il KP memorizza i numeri seriali dei dispositivi, il loro indirizzo MAC, il PIN necessario per l'accoppiamento Bluetooth e informazioni sul protocollo da usare per il dispositivo; inoltre viene creata una sottoscrizione che notifica il KP quando nuovi dispositivi medici sono associati all'utente o rimossi, in modo da avere la tabella dei dispositivi associati all'utente sempre aggiornata. A run-time quando il KP scopre uno di questi dispositivi, quest'ultimo viene automaticamente riconosciuto; viene effettuato il pairing, applicato il protocollo corretto e quindi effettuato il parsing dei dati i quali vengono pubblicati nella SIB e associati al

paziente. Nella figura 11 è mostrata parte del sottografo presente nella SIB e utilizzata dal KP per realizzare questa funzionalità.

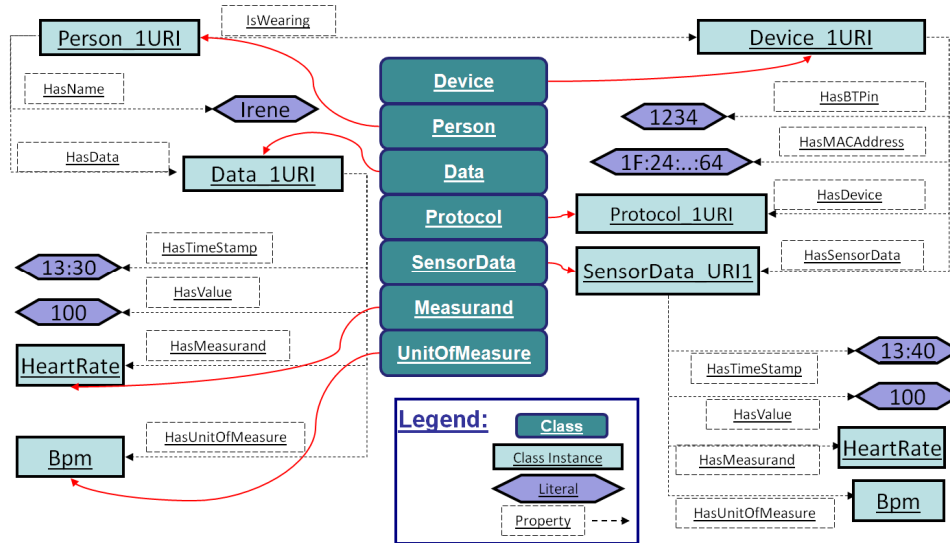


Figura 12 Sottografo utilizzato dal Physiological Sensors KP

ThomIndex KP

Questo KP è responsabile di calcolare l'indice termoisometrico per ogni ambiente del sistema. E' un KP aggregatore. Il KP si sottoscrive ai dati di temperatura e umidità di una stanza ed effettua dei controlli sulle unità di misura; ad ogni notifica ricevuta da parte della SIB viene calcolato il nuovo indice di Thom ed effettuato un update. L'informazione è associata all'ambiente che contiene i sensori. Il KP calcola l'indice di Thom usando la temperatura e l'umidità media a partire dai dati forniti dai sensori. Ogni stanza ha un KP di questo tipo, realizzato per comodità in Python.

Location KP

Questo KP è un *legacy adapter* che si interfaccia all'RFID reader collocato all'ingresso di una stanza. Nel nostro scenario ogni reader è associato durante l'inizializzazione ad una stanza; quando una persona entra nella stanza e pone il suo RFID tag nelle vicinanze del reader il KP aggiorna la posizione dell'utente. Il KP aggiorna anche la posizione dei dispositivi indossati dalla

persona. Un *Location KP* è necessario per ogni ambiente e RFID reader. Il KP gira su un PC connesso allo SS.

AlarmGenerator KP

Questo KP effettua una query per cercare tutte le entità che hanno un *SafetyLimitOnParameter* associato. Quindi scaricati i valori in una tabella il KP si sottoscrive ai dati di interesse. Quando il dato monitorato è fuori soglia, viene creata un'istanza di *Alarm*, pubblicata nella SIB e associata all'entità pertinente.

ManualAlarmGenerator

Questo KP permette allo staff medico di creare manualmente delle istanze di *Alarm* in caso di situazioni anomale.

Health Care Monitoring KP

Questo KP consente allo staff medico di selezionare l'utente da monitorare tra quelli disponibili; quindi viene creata una sottoscrizione su tutti i dati rilevanti. Tutte le istanze di *Data* associate alla persona sono utilizzate, non solamente i dati quelli fisiologici.

Il KP visualizza quando disponibili i valori istantanei di heart rate, temperatura della pelle, frequenza respiratoria, angolo di postura e un indice di attività; visualizza inoltre le ultime misure effettuate da bilancia, sfigmomanometro e pulsossimetro. Al tempo stesso è mostrata la posizione in cui si trova l'utente monitorato e le condizioni ambientali (temperatura, umidità, indice di Thom). Questo KP è scritto in Python e lo stesso software può essere utilizzato sia su un PC Ubuntu sia su smartphone Maemo Nokia N900.

Alarm Dispatcher KP

Questo KP si sottoscrive alle creazioni di istanze di *Alarm* e quando è notificato invia una mail o un sms con le relative informazioni al medico. Sono stati realizzati due metodi alternativi per inviare SMS; il primo sfrutta un KP sullo smartphone Nokia N900 ed il secondo un KP che gira su un PC collegato ad un modulo Siemens HC25.

History KP

Il KP si sottoscrive alle variazioni dei parametri dell'utente effettuando uno storico dei dati monitorati; nella versione attuale i dati sono raccolti con il loro timestamp in un file.

In figura 12 è mostrato uno screenshot dell' Health Care Monitoring; l'utente monitorato è Irene e lo smile verde indica che non sono presenti allarmi legati al paziente. Nella colonna di sinistra è indicata il nome della stanza in cui si trova l'utente (in questo caso cellar) e le informazioni ambientali. Sulla destra sono presenti le informazioni fisiologiche.

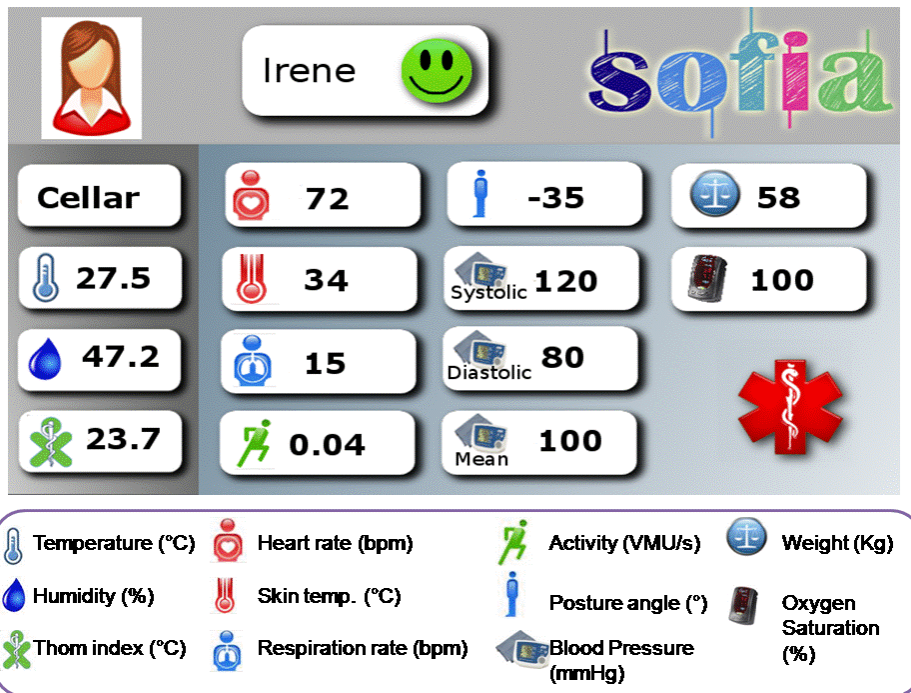


Figura 13 Healthcare monitoring KP screenshot

In figura 13 sono mostrati i dati raccolti durante una sessione di monitoraggio simulata in laboratorio durante la quale l'utente si spostava ripetutamente tra due stanze indossando lo Zephyr BioHarness.

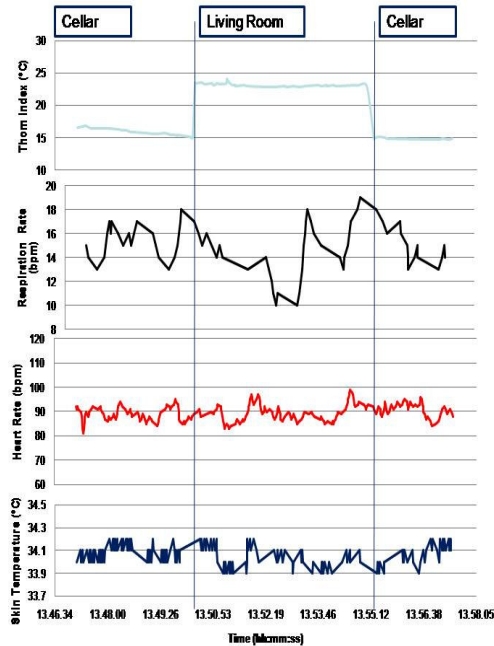


Figura 14 Esempio di una sessione di monitoraggio

7.3 Conclusioni

In questo capitolo è stata presentata il design e la realizzazione di un'applicazione di monitoraggio concorrente di parametri fisiologici e ambientali basata su un sistema interoperabile. Il risultato è un prototipo SS che decisamente aumenta l'interoperabilità rispetto un sistema standard come quello mostrato nel capitolo 4. Questo è stato dimostrato integrando diversi KP, produttori, consumatori e aggregatori di informazione condivisa. Tutti i KP dell'applicazione sono stati scritti lavorando a livello di SSAP e di triple; il problema è che usando erroneamente l'ontologia si può precludere il corretto funzionamento del sistema. Per questo è stato sviluppato all'interno del progetto SOFIA un Application Development Kit (ADK); questo tool è un generatore di librerie a partire dall'ontologia che ha lo scopo di facilitare il compito dello sviluppatore. Un altro tool molto utile in fase di realizzazione e attualmente in sviluppo è un visualizzatore/editor grafico per facilitare l'esplorazione e l'aggiunta di informazioni nello SS.

8 Conclusioni

L'esperienza maturata nel progetto basato su Smart Space di sistemi di monitoraggio di parametri ambientali e fisiologici ci consente di formulare una serie di raccomandazioni. Lo scopo di questa metodologie progettuale è aiutare il designer nel realizzare una sistema per condividere l'informazione in modo interoperabile tra dispositivi e applicazioni eterogenee. Attraverso il disaccoppiamento di KP e la capacità da parte della SIB di condividere diverse ontologie, i sistemi basati su SS possono crescere integrando nuovi dispositivi e applicazioni. L'abilità di scambiare e condividere dati e informazioni è la caratteristica dello Smart Space che permette di realizzare applicazioni innovative.

8.1 Raccomandazioni per lo sviluppo di un sistema interoperabile

Progettare un sistema interoperabile può essere un lavoro difficile da intraprendere senza una linea guida. Le seguenti raccomandazioni per la realizzazione di sistemi interoperabili basati su Smart Space saranno utili sia a sviluppatori che vogliono ottenere interoperabilità tra dispositivi e sistemi legacy realizzando applicazioni e software innovativo, sia a progettisti che intendono realizzare sistemi in grado di produrre e consumare direttamente informazioni interoperabili.

Utilizzare l'RDF per descrivere i dati del sistema e seguire le raccomandazioni del W3C.

Il Resource Description Framework (RDF) è stato introdotto dal W3C per lo scambio di informazioni sul il web semantico. Questo significa che milioni di sistemi condivideranno i dati in RDF. Inoltre molte comunità di sviluppatori offrono tools, sistemi di archiviazione e linguaggi di query che fanno dell'RDF un robusto framework per modellare li informazione. L'RDF permette di rag-

giungere interoperabilità delle informazioni anche in sistemi in cui l'informazione è estremamente dinamica come in telemedicina.

Utilizzare ontologie descritte in OWL per dare una struttura e un significato semantico all'informazione.

L'ontologia infatti consente di assegnare una struttura semantica all'informazione che è espressa in un grafo RDF verrebbe a mancare. Le ontologie svolgono un ruolo fondamentale in quanto assegnano un preciso significato ai dati rappresentati in RDF. Questo aspetto è vitale laddove sia condivisa l'informazione in quanto consente a diversi sistemi di lavorare con lo stesso vocabolario. L'OWL è lo standard riconosciuto e proposto da W3C e sono disponibili molti tools per modellare ontologie ed in grado di fornire strumenti di reasoning. Per lo stesso motivo e per incoraggiare interoperabilità tra sistemi, un elevato numero di ontologie è pubblicato liberamente sul web.

Minimizzare il numero di comunicazioni eterogenee utilizzate nel sistema.

Questo requisito viene dall'esigenza di avere interoperabilità a livello di comunicazione. Maggiore è il numero di tecnologie di comunicazione impiegate (ad es. WiFi, Bluetooth, ZigBee, ..) maggiore è il tempo e l'effort richiesto per sviluppare il sistema. In ogni caso l'approccio basato su SS richiede unicamente di implementare la comunicazione tra dispositivi e SIB e non tra ogni coppia di dispositivi.

Non trascurare la gestione delle eccezioni.

Si ricordi che nell'approccio basato su SS la verifica della consistenza dell'informazione è lasciata all'applicazione. Ad esempio è necessario controllare la disponibilità dei dati nello SS. Non ci sono garanzie che l'informazione di cui necessita un KP sia presente nello SS proprio per la dinamicità del sistema; il KP che utilizza l'informazione deve prevedere la possibilità che un'informazione non sia presente.

I dispositivi e le applicazioni devono portare con se la parte di ontologia che utilizzeranno e condividerla nello SS.

Infatti quando uno SS è inizializzato non contiene nessun tipo di informazioni. Ci sono quindi due opzioni. La prima prevede di inserire tutta l'ontologia

come passo iniziale; ovvero un KP dedicato si occupa di inserire tutta l'ontologia nella SIB. La seconda opzione consiste nel far inserire ad ogni KP la parte di ontologia che utilizzerà.

Nel prototipo discusso precedentemente abbiamo adottato la prima soluzione.

Lo scambio di informazioni deve avvenire sempre attraverso la SIB.

L'interoperabilità è assicurata condividendo e scambiando l'informazione attraverso lo SS e interpretata grazie all'ontologia e disaccoppiando produttori e consumatori. Una tripla RDF rappresenta una affermazione riguardo lo stato delle risorse come per esempio il valore corrente di un dato o la posizione in cui si trova una persona. Viceversa se il sistema necessita di ulteriori dati come per esempio file o collegamenti a database esterni la SIB risulta nuovamente idonea per essere integrata con questi sistemi. Infatti la SIB gestisce solamente informazione ma è possibile fare riferimento a informazioni esterne tramite URI. Questa caratteristica è molto potente per integrare al sistema repository e risorse esterne.

Utilizzare le sottoscrizioni liberamente.

L'approccio basato su SS consente di applicare un modello di notifica publish/subscribe. Le sottoscrizioni sono un potente meccanismo per notificare eventi le applicazioni interessate. L'utilizzo delle sottoscrizioni massimizza la reattività del KP mantenendo la sua informazione locale sempre aggiornata rispetto la SIB.

Le informazioni nello SS rappresentano una descrizione dello stato attuale; conviene gestire i dati storici se necessari con KP ad hoc.

Lo SS deve rappresentare lo stato corrente di un ambiente fisico essendone la rappresentazione digitale. KP dedicati possono sottoscrivere alle informazioni rilevanti e gestirle nel modo più appropriato e in particolare possono costruire archivi storici.

Le ontologie devono essere condivise e riutilizzate per massimizzare l'interoperabilità.

L'uso di un'ontologia condivisa è fondamentale per mantenere l'informazione interoperabile. Per quanto sia possibile realizzare KP traduttori

di ontologie, la miglior pratica è utilizzare ontologie comuni per integrare sistemi diversi senza nessun costo aggiuntivo.

Assumere che le informazioni prodotte saranno riutilizzate e quindi documentare l'ontologia per permettere a futuri sviluppatori di capire la semantica dei dati.

L'ontologia deve essere documentata e completata da esempi che ne illustrino l' utilizzo; questi riguarderanno la notazione usata per creare URI, il corretto utilizzo delle proprietà e esempi di istanze complete.

Utilizzare un class ontology based library generator per astrarre il progettista dall'ontologia.

Può essere interessante sviluppare una applicazione senza dover accedere all'ontologia. Questo può essere realizzato tramite un tool che genera classi a partire dal file OWL contenente l'ontologia. Questo tool fornisce allo sviluppatore la possibilità di lavorare con classi standard senza dover lavorare a livello di SSAP, ontologie ed RDF.

Queste considerazioni nascono dall'esperienza maturata durante il dottorato e durante l'attività di progetto discussa nei capitoli 3, 4, 5, 6 e 7.

Il progetto descritto nel capitolo 3 è stato realizzato per monitorare il segnale ECG e il contesto dell'utente per poter capire la correlazioni tra anomalie cardiache e condizioni ambientali. Per realizzare il sistema è stato necessario integrare fra loro un elevato numero di sensori per la ricostruzione del contesto. L'utilizzo della piattaforma Java "Mobicomp" come *context repository* ha dato la possibilità di memorizzare l'informazione e di garantirne l' accesso; la mancanza di un'ontologia ha portato a forti limitazioni. Senza ontologia qualsiasi tipo di informazione può essere condivisa ma il consumatore deve sapere autonomamente come interpretare l'informazione. La decisione di realizzare un sistema integrato in grado di acquisire tutte le informazioni era stata motivata dall'idea che fosse più semplice rispetto a integrare fra loro differenti sistemi. In realtà l'uso di uno Smart Space ha permesso di realizzare questa integrazione e ha concesso molti benefici.

Utilizzando il prototipo di ontologia discusso nel capitolo 5 e sfruttando Smart-M3 per creare uno SS, l'usabilità e i vantaggi di questo tipo di sistemi sono risultati ben evidenti. La possibilità di aggiungere sensori e integrare differenti tipi di dati, la realizzazione di applicazioni per visualizzare informazioni e la creazione di aggregatori sono stati facilitati dall'utilizzo di una ontologia condivisa. Il prototipo presentato infatti è l'evoluzione di un primo sistema molto più semplice; man mano abbiamo aggiunto sensori e funzionalità, spesso traendo beneficio dal riutilizzo di codice, e senza dover modificare moduli già esistenti. I vantaggi offerti dal sistema sono stati evidenziati riutilizzando e estendendo il prototipo di ontologia per realizzare uno scenario di manutenzione all'interno del progetto SOFIA.

8.2 Sviluppi futuri

Nella tesi è stato discusso la metodologia per lo sviluppo e la realizzazione di uno SS per la realizzazione di un sistema di monitoraggio innovativo. È stato utilizzato il prototipo fornito da Nokia di Smart-M3 per mostrare le opportunità e le funzionalità offerte da una soluzione di questo tipo. Il sistema Smart-M3 sta diventando maturo e l'Application Development Kit (ADK) faciliterà il lavoro di sviluppatori. Per questi motivi la piattaforma sarà integrata in un sistema più ampio cioè all'interno dell'architettura di CHIRON. Il progetto CHIRON ha accolto la piattaforma come nucleo per generare interoperabilità tra fonti eterogenee; questo grazie alle funzionalità che uno SS offre sia per quanto riguarda la possibilità di offrire un modo semplice per realizzare un profilo personalizzato sia per il modo di fornire informazioni ad agenti esterni quali motori a regole. Con questa ipotesi l'architettura di CHIRON può essere rivista come mostrata in figura 15 utilizzando i concetti di SS.

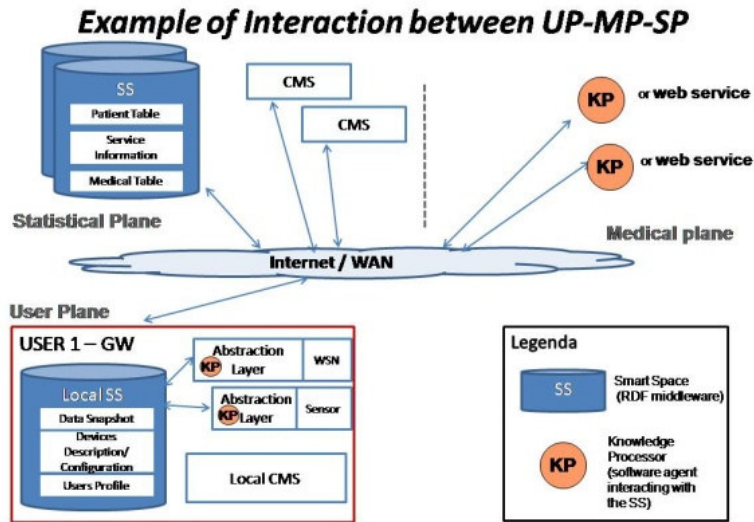


Figura 15 Esempio di interazione tra user medical e statistical plane

Uno Smart Space può essere creato per ogni livello di utenza (*user / medical / statistical plane*). Nel piano utente, una SIB locale supporta la memorizzazione del contesto corrente (profilo dell'utente e descrizione/configurazione dei dispositivi – *Alter Ego*). Gli *abstraction layers* allineano le informazioni dei dati grezzi con l'ontologia di riferimento usata dalla SIB. Un *content manager system* locale e referenziato nella SIB può collezionare i dati storici a breve termine. Il piano medico monitora i dati dell'utente e modifica informazioni o sull'utente o sulla configurazione dei dispositivi; l'accesso può avvenire direttamente da KP o da web service. I dati storici utili al fine statistico sono memorizzati tramite specifici KP in CMS o DB esterni, sempre referenziati in SS esterni. In questa situazione essendo un middleware per la raccolta e condivisione di dati medici un'opportuna scelta è quella di utilizzare un'ontologia mappata a partire dai vocabolari e standard medici quali HL7.

L'approccio utilizzato apre a nuovi scenari nella ricerca scientifica in telemedicina. Ad esempio si aprono percorsi verso la realizzazione di nuovi servizi e applicazioni che consentono da un lato di studiare la correlazione tra i parametri raccolti durante l'osservazione di determinati fenomeni, dall'altro di rilevare situazioni anomale attraverso l'implementazione di algoritmi predittivi dopo aver istruito il sistema con tecniche di learning.

Si ritiene che il più interessante risultato del Dottorato sia proprio lo studio dei modelli di utilizzo delle piattaforme interoperabili e la proposta e successiva adozione di detto modello in un importante progetto di ricerca europeo sulla

telemedicina a cui partecipano importanti player del settore, tra cui FIMI PHILIPS, PHILIPS HEALTHCARE, ALMA IT SYSTEM, I+, SOUTHAMPTON UNIV. HOSPITAL, CARDIOCINETICS, BARCO e ZORGGEMAK.

Bibliografia

- [1] A. Bonfiglio, S. Cerutti, S. De Rossi, G. Magenes, Sistemi intelligenti per la salute e protezione dell'uomo, Pàtron Editore
- [2] Population Project EUROPOP, Available: <http://epp.eurostat.ec.europa.eu>
- [3] Institute for Healthcare Improvement (IHI), Available: <http://www.ihl.org/>
- [4] Deloitte&Touche, The Emerging European Health Telematics Industry, Health Information Society Technology Based Industry Study, on assignment of European Commission-Directorate General Information Society, 25/4/2000
- [5] J. Walker, E. Pan, D. Johnston, J. Adler-Milstein, D. W. Bates, B. Middleton, The Value Of Health Care Information Exchange And Interoperability, Health Affairs – Web Exclusive, published ahead of print January 19, 2005, doi:10.1377/hlthaff.w5.10
- [6] J. Park, S. Ram, Information systems interoperability: What lies beneath?, in ACM transactions on information systems, vol. 22, no 4, Oct. 2004, pp. 595-632
- [7] L. S Kalkstein; K. M. Valimont. Climate effects on human health. Technical report, U.S. Environmental Protection Agency, 1987
- [8] R. Mark, Telemedicine system: the missing link between homes and hospitals?, Mod Nurs Home, 1974 Feb, 32(2), pp. 39-42
- [9] J. P. Finley, D. G. Human, M. A. Nanton, D. L. Roy, R. G. Macdonald, D. R. Marr, H Chiasson, Echocardiography by telephone-- Evaluation of pediatric heart disease at a distance, The American Journal of Cardiology, Volume 63, Issue 20, 15 June 1989, Pages

1475-1477, ISSN 0002-9149, DOI: 10.1016/0002-9149(89)90011-8

- [10] K. M. Zundel, Telemedicine: history, applications, and impact on librarianship, *Bull Med Libr Assoc.* 1996 January; 84(1): 71–79
- [11] OLDES (Older people's e-services at home), Available: <http://www.oides.eu/>
- [12] N. Noury, AILISA: experimental platforms to evaluate remote care and assistive technologies in gerontology, *IEEE Proceedings of the 7th International Workshop on Enterprise Networking and Computing in Healthcare Industry*, Busan, South Korea, June 23-25, 2005, pp. 67-72
- [13] P. Steggles, S. Gschwind, The Ubisense Smart Space Platform, *Advances in Pervasive - Computing*, Adjunct Proceedings of the Third International Conference on Pervasive Computing. Vol. 191, pp 73-76, 2005
- [14] L. Jiang, D. Liu, B. Yang, Smart home research, *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, Vol. 2, pp. 659- 663, August 2004
- [15] N. Noury, G. Virone, P. Barralon, J. Ye, V. Rialle, J. Demongeot, New trends in health smart homes, *Proceedings of the 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (Healthcom 2003)*, pp. 118 – 127, June 2003
- [16] Mainstreaming on Ambient Intelligence (MonAMI) project, Available: <http://www.monami.info>
- [17] Smart Medical Home Research Laboratory, Center for Future Health, University of Rochester, Available: <http://www.futurehealth.rochester.edu>
- [18] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner, W. Newstetter, The Aware Home: A Living Laboratory for Ubiquitous Computing Research, *Proceedings of the*

- 2nd International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture, pp. 191-198, October 1999
- [19] F. Knoefel, V. Emerson, B. Schulman, TAFETA: An Inclusive Design for Tele-Health, Technology and Persons with Disabilities Conference, March 2005
- [20] S. Helal, W., Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, The Gator Tech Smart House: A Programmable Pervasive Space, IEEE Computer, Vol. 38, No. 3, pp. 50-60, March 2005
- [21] Home Depot Smart Home, Pratt School of Engineering, Duke University, Available: <http://www.smarthome.duke.edu>
- [22] P. Bartolomeu, J. Fonseca, V. Santos, A. Mota, V. Silva, M. Sizeando, Automating Home Appliances for Elderly and Impaired People: The B-Live Approach, Proceedings of the 2nd International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI2007), November 2007
- [23] T. Winograd, Architectures for context. Human-Computer Interaction, 16(2-4):401-419, Dec. 2001
- [24] B. Schilit, N. Adams, and R. Want, Context-aware computing applications, In IEEE Workshop on Mobile Computing Systems and Applications, SantaCruz, CA, US, 1994. Available from World Wide Web: citeseer.ist.psu.edu/schilit94contextaware.html
- [25] G. Chen and D. Kotz, A survey of context-aware mobile computing research, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000
- [26] A.K. Dey and G.D. Abowd, Towards a better understanding of context and contextawareness, Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, June 1999
- [27] M. Korkea-aho, Context-aware applications survey. In Internet-working Seminar (Tik-110.551), Helsinki, Finland, 2000

- [28] G. Chen and D. Kotz. A survey of context-aware mobile computing research, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000
- [29] M. Weiser, The Computer for the 21st Century, *Scientific Am.*, Sept., 1991, pp. 94-104; reprinted in *IEEE Pervasive Computing*, Jan.-Mar. 2002, pp. 19-25
- [30] L. L. Brownsword, D. J. Carney, D. Fisher, G. Lewis, C. Meyers, Current perspectives on interoperability, Technical Report CMU/SEI-2004-TR-009 CMU/SEI-2004-TR-009, Carnegie-Mellon University, March, 2004
- [31] IEEE Standards Committee. Ieee std 610.12-1990 ieee standard glossary of software engineering terminology. Available: http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html
- [32] A. Lappeteläinen, J-M. Tuupola, A. Palin and T. Eriksson, Networked systems, services and information The ultimate digital convergence, 1st International NoTA Conference, 11th June 2008, Helsinki, Finland
- [33] H. Zimmerman, OSI reference model — The ISO model of architecture for open systems interconnection, *IEEE Transactions on Communications*, 28, Apr. 1980, pp. 425–432
- [34] S. Blackburn, *The oxford dictionary of philosophy* electronic resource/ simon blackburn, 2008
- [35] T. R. Gruber, Towards Principles for the Design of Ontologies Used for Knowledge Sharing, In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers. Available from World Wide Web: citeseer.ist.psu.edu/gruber93toward.html
- [36] P. Ohrstrom, J. Andersen, H. Scharfe, What has happened to ontology, In Frithjof Dau, Marie-Laure Mugnier, and Gerd Stumme,

- editors, Proceedings of the 13th International Conference on Conceptual Structures (ICCS 2005), volume 3596 of Lecture Notes in Computer Science, pages 425-438. Springer, 2005.
- [37] J. Heflin, OWL web ontology language use cases and requirements. W3C recommendation, W3C, February 2004.<http://www.w3.org/TR/2004/REC-webont-req-20040210/>
- [38] B. Chandrasekaran, J. R. Josephson, and V. Richard Benjamins, What are ontologies, and why do we need them?, IEEE Intelligent Systems, 14:20-26, 1999
- [39] C. Welty, D. L. McGuinness, and M. K. Smith, OWL web ontology language guide, W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
- [40] D. Cook and S. Das, Smart Environments: Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing). Wiley-Interscience, 2004
- [41] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, Imote2: Serious computation at the edge, Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International, pages 1118-1123, Aug. 2008
- [42] R. Adler, M. Flanigan, J. Huang, R. Kling, N. Kushalnagar, L. Nachman, C. Wan, and M. Yarvis, Intel mote 2: an advanced platform for demanding sensor network applications, In SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems, pages 298-298, New York, NY, USA, 2005. ACM
- [43] Taos. Tsl2561, 2009. Available from World Wide Web: <http://www.taosinc.com/downloaddetail.aspx?did=140>
- [44] Sensirion. Sht15 temperature and humidity sensor, 2009. Available from World Wide Web: http://www.sensirion.com/en/01_humidity_sensors/03_humidity_sensor_sht15.htm

- [45] 45ST Microelectronics. Lis3102dq, 2008. Available from World Wide Web: <http://www.st.com/stonline/books/pdf/docs/10175.pdf>
- [46] G. D. Clifford, F. Azuaje, and P. McSharry, Advanced Methods And Tools for ECG Data Analysis, Artech House, Inc., Norwood, MA,USA, 2006
- [47] E.M. Spinelli, R. Pallas-Areny, and M.A. Mayosky, Ac-coupled front-end for biopotential measurements, Biomedical Engineering, IEEE Transactions on, 50(3):391-395, Mar 2003
- [48] J. Pan and W. J. Tompkins, A real-time qrs detection algorithm, Biomedical Engineering, IEEE Transactions on, BME-32(3):230-236, March 1985
- [49] I Christov, Real time electrocardiogram qrs detection using combined adaptive threshold, BioMedical Engineering OnLine, 3(1):28, 2004
- [50] W. Kyle, The human bioclimate of Hong Kong, Brazil Rand M Kolar Eds Proceedings of the Contemporary Climatology Conference 1994;345–350
- [51] World Wide Web Consortium – W3C <http://www.w3.org/>
- [52] J. J. Carroll and G. Klyne, Resource description framework (RDF): Concepts and abstract syntax, W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- [53] D. Beckett, RDF/xml syntax specification (revised). W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-rdf-syntaxgrammar-20040210/>
- [54] T. Berners-Lee. Notation 3, Notation 3. Available from World Wide Web: <http://www.w3.org/DesignIssues/Notation3.html>
- [55] D. Becket, T. Berners-Lee, Turtle - terse rdf triple language. Team submission, W3C, January 2008. Available from World Wide Web: <http://www.w3.org/TeamSubmission/turtle/>

- [56] R. V. Guha and D. Brickley, RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- [57] G. Schreiber and M. Dean, OWL web ontology language reference. W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
- [58] L. Ding, P. Kolari, Z. Ding, S. Avancha, T. Finin, A. J. L. Ding, P. Kolari, Z. Ding, S. Avancha, T. Finin, A. Joshi, Using ontologies in the semantic web: A survey. Technical report, University of Maryland, July 2005
- [59] M. Minsky, A framework for representing knowledge. Technical report, Cambridge, MA, USA, 1974
- [60] C. Welty, D. L. McGuinness, and M. K. Smith, OWL web ontology language guide. W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
- [61] O. Lassila, Programming Semantic Web Applications: A Synthesis of Knowledge Representation and Semi-Structured Data. PhD thesis, Helsinki University of Technology, November 2007
- [62] I. Niles and A. Pease, Origins of the ieee standard upper ontology, In Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology, pages 4-10, 2001
- [63] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, R. Oltramari, L. Schneider, L. Partner and I. Horrocks, Wonderweb deliverable d17, the wonderweb library of foundational ontologies and the dolce ontology, 2002
- [64] D.J. Russomanno, C. Kothari, and O. Thomas, Building a sensor ontology: A practical approach leveraging iso and ogc models. In The 2005 International Conference on Artificial Intelligence, 2005
- [65] M. Botts and A. Robin, Opengis R sensor model language (sensorml) implementation specification. OpenGIS Implementation

Specification OGC 07-000, Open Geospatial Consortium Inc., 07 2007. Version:1.0.0.

- [66] International Organization for Standardization. Geographic information metadata. Technical report, 2010, 2010. Available from World Wide Web: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020
- [67] M. Eid, R. Liscano, and A. El Saddik, A universal ontology for sensor networks data, In IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2007. CIMSA 2007, pages 59-62, 2007
- [68] J. Honkola, H. Laine, R. Brown, O. Tyrkko, Smart-M3 information sharing platform, Computers and Communications (ISCC), 2010 IEEE Symposium on , vol., no., pp.1041-1046, 22-25 June 2010
- [69] Smart-M3, Wikipedia, Available: <http://en.wikipedia.org/wiki/Smart-M3>
- [70] Smart-M3 public source code, Available : <http://sourceforge.net/projects/smart-m3/>
- [71] SOFIA (Smart Object For Intelligence Applications), Available: <http://www.sofia-project.eu/>
- [72] X. Wang, J. Song Dong, C. Chin, S. Hettiarachchi, and D. Zhang, Semantic space: An infrastructure for smart spaces, IEEE Pervasive Computing, 3:32-39, 2004
- [73] SOFIA Deliverable 5.11: Interoperability Platform Principles, Available : <http://www.sofia-project.eu/node/329>
- [74] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke, Ontoedit: Collaborative ontology development for the semantic web, In ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web, pages 221-235, London, UK, 2002. Springer-Verlag.] [P. Lambrix, M. Habbouche,

- and M. Perez. Evaluation of ontology development tools in bioinformatics, 2003
- [75] M. F. Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra, Building a chemical ontology using methontology and the ontology design environment, *IEEE Intelligent Systems*, 14:37-46,1999
- [76] H. Mizen, C. Dolbear, and G. Hart, Ontology ontogeny: Understanding how an ontology is created and developed. In M. Andrea Rodriguez, Isabel F. Cruz, Max J. Egenhofer, and Sergei Levashkin, editors, *GeoS*, volume 3799 of *Lecture Notes in Computer Science*, pages 15-29. Springer, 2005
- [77] N. F. Noy and D. L. McGuinness, *Ontology development 101: A guide to creating your first ontology*, Technical report, 2001
- [78] CHIRON (Cyclic and person-centric Health management: Integrated appRoach for hOme, mobile and clinical eNvironments) Project. Available: <http://www.chiron-project.eu>
- [79] Eurotech group, 2009. Available from World Wide Web: <http://www.eurotech.com/>
- [80] F. Vergari, S. Bartolini, F. Spadini, A. D'Elia, G. Zamagni, L. Roffia, T.S. Cinotti, A Smart Space application to dynamically relate medical and environmental information, *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010 , vol., no., pp.1542-1547, 8-12 March 2010
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5457056&isnumber=5456897>
- [81] F. Spadini, F. Vergari, L. Nachman, C. Lamberti, T.S Cinotti, A wireless and context-aware ECG monitor: An iMote2 based portable system, *Computers in Cardiology*, 2008 , vol., no., pp.997-1000, 14-17 Sept. 2008
doi: 10.1109/CIC.2008.4749212
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4749212&isnumber=4748952>

- [82] A. D'Elia, L. Roffia, G. Zamagni, F. Vergari, A. Toninelli, P. Bellavista, Smart applications for the maintenance of large buildings: How to achieve ontology-based interoperability at the information level, Computers and Communications (ISCC), 2010 IEEE Symposium on , vol., no., pp.1-6, 22-25 June 2010 doi: 10.1109/ISCC.2010.5546633 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5546633&isnumber=5546495>
- [83] F. Vergari, V. Auteri, C. Corsi, C. Lamberti, A ZigBee-based ECG transmission for a low cost solution in home care services delivery, MESPE Journal (Mediterranean Journal of Pacing and Electrophysiology) 2007; 3-4:181-186URL: <http://www.mespe.net/en/newselem/20080323172650/>
- [84] L. Roffia, A. D'Elia, F. Vergari, D. Manzaroli, S. Bartolini, G. Zamagni, T. Salmon Cinotti, J. Honkola, A Smart-M3 lab course: approach and design style to support student projects", 8th FRUCT Conference, Lappeenranta, Finland, 9-12 November 2010
- [85] F. Spadini, F. Vergari, G. Zamagni, S. Bartolini, R. Trevisan, A. D'Elia, L. Roffia, D. Manzaroli, C. Lamberti, T. Salmon Cinotti, J. Honkola, H.E. Laine, Abstracting Knowledge from Physical Parameters in Smart Spaces: An M3 Demonstration, NoTA Conference 2009 , Sept.30th – Oct. 1st San Jose CA, Poster 9.
- [86] T. Cinotti Salmon, S. Bartolini, A. D'Elia, L. Roffia, F. Spadini, R. Trevisan, F. Vergari, G. Zamagni, Approaching the design of interoperable smart environments applications. NoTA 2009 Conference
- [87] C. Lamberti, F. Vergari, Telemedicine –Technical Aspects, MESPE Journal (Mediterranean Journal of Pacing and Electrophysiology) 2007; 3-4:125-126 D.
- [88] F. Vergari, T. Salmon Cinotti, A. D'Elia, L. Roffia, G. Zamagni, C. Lamberti, An integrated framework to achieve interoperability in person-centric health management, Submitted to: International Journal of Telemedicine and Applications

