THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *12/12/2017* **par :**

Rita ZGHEIB

SeMoM: A semantic Middleware for IoT Healthcare Applications

JURY

Myriam LAMOLLE	Professeure-Université de Paris 8	Rapporteur
Norbert NOURY	Professeur-Université de Lyon	Rapporteur
Lynda TAMINE	Professeure-Université de Toulouse3	Examinateur
Ernesto EXPOSITO	Professeur-Université de Pau	Examinateur
Rémi BASTIDE	Professeur-INU Champollion	Directeur de thèse
Emmanuel CONCHON	MCF-Université de Limoges	Codirecteur de thèse

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)

Directeur(s) de Thèse :
Rémi BASTIDE et *Emmanuel CONCHON*

Rapporteurs :
Myriam LAMOLLE et *Norbert NOURY*

Acknowledgements

First, I would like to thank the “Midi-Pyrenees Region" and the "CUFR Jean-François Champollion” for funding my PhD thesis. Without such a grant this work would not have been realized.

I would like to express my sincere gratitude to my thesis director Prof. Rémi Bastide for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance increased the productivity of my research and helped me in all the time of research and writing of this thesis.

A big thanks to Dr. Emmanuel Conchon for his guidance, recommendations, follow-up and his constant support throughout my Ph.D. I am lucky to have you as a co-director of my thesis, our brainstorming and continuous discussions were a source of inspiration. Thank you for challenging me and helping me to develop my ideas.

My deep gratitude is also addressed to the members of jury: Prof. Myriam Lamolle and Prof. Norbert Noury for reviewing my dissertation and providing valuable comments. Thanks to Prof. Lynda Tamine and Prof. Ernesto Exposito for accepting being part of my PhD committee members.

Thanks to all people of ENEA research center for their welcome and hospitality in Rome. I really enjoyed being there and have fruitful discussions with you. A special thank to Antonio De Nicola and Maria Luisa Villani for the work and the support during my time in Rome, which helped shaping many aspects of this thesis.

I would also like to thank Pr. Thomas Plagemann, Pr. Vera Goebel, Dr Stein kristiansen from the university of Oslo for the collaboration and work in the beautiful city Oslo. Thanks to all DMMS PhD students for the sightseeing tour in Oslo, it was perfect.

I have the pleasure of thanking "ISIS" team members for the moments that we spent together and exchanged ideas. Thanks to Dr Jing, Dr. Nicolas Singer, Dr Nathalie Souf, Dr. Elyes Lamine, Dr. Imen Megdiche, Dr. Adrien Deffosez, Prof. Hérvé Pingaud, Dr. Rejane Dalce, Prof. Bernard Rigaud, Amal and Rafika for the discussions about my thesis and about computer science in general. I would also like to thank all my colleagues Samuel, Florence, Eric, Véronique, Laurent, Lauriane, Alexandre and Emmanuelle for the discussions and enjoyable launch breaks. A special thanks to Bruno Roussel for his help in the CHL and to Francis Faux to the friendship and the ping pong breaks all these three years.

In my daily work I have been blessed with a friendly and cheerful group. A big thanks to my colleague Ghazar for sharing the office, for the enjoyable discussions and breaks. I extend my grateful to Marine who has become a close friend to me, thank you for your support, funny discussions and the new French expressions! Thank you copineee Fiona to all our funny moments and to Anne for the Salsa nights, you made the stay in Castres very enjoyable. I appreciate the support from all of you and the strong friendship that has grew along the Ph.D.

I would like to thank my friends in the "JeunesPro" group in Castres: Annelise, Mylène, Maud, Emilie, Hélène, Marie-Pauline, Caroline, Charlotte, Ombelline, Bérengère, Jean-Baptiste et père Philippe. Thank you for sharing pleasant moments, the convivial evenings and the shared meals allowed me to better know the French culture and to better integrate into society. I would like to thank Brigitte and the "Secours Catholique Castres" members with whom I shared also pleasant moments and I realized human activities that I always liked to do in my life.

I would like to thank my Libano-Toulousains friends Batoul, Vita, Aude, Anthony, Laura, Mouin, Rana, Dany, Amar, Ali, Raghda, Farid, Arwa, Guevorg, Haroutine and all "Les amis du Levant" organization. Cheers to all of you, as we always say: you are my family in France.

None of this would have been possible without the support and love of my beautiful family. I am blessed by the prayers and candles of my mother Thérèse (Terro), the advice and care of my father Elias (Yeso), the daily encouragement of my sisters and brother Marie, Ahlam, Hiam and Simon. They all kept me going higher, and pushed me to be the perfect person I can be. Thank you !

Finally, I would like to thank my fiancé Jocelyn. He inspired me by his love and addiction to computer science and technology. I am lucky to have a caring person like you beside me. You pushed me to realize all my dreams and do not worry about anything else. Thank you!

Abstract

Nowadays, the adoption of the Internet of Things (IoT) has received a considerable interest from both academia and industry. It provides enhancements in quality of life, business growth and efficiency in multiple domains. However, the heterogeneity of the “Things” that can be connected in such environments makes interoperability among them a challenging problem. Moreover, the observations produced by these “Things” are made available with heterogeneous vocabularies and data formats. This heterogeneity prevents generic solutions from being adopted on a global scale and makes difficult to share and reuse data for other purposes than those for which they were originally set up. In this thesis, we address these challenges in the context of healthcare applications considering how we transform raw data to cognitive knowledge and ontology-based information shared between IoT system components.

With respect to heterogeneity and integration challenges, our main contribution is an ontology-based IoT architecture allowing the deployment of semantic IoT applications. This approach allows sharing of sensors observations, contextualization of data and reusability of knowledge and processed information. Specific contributions include:

- Design of the Cognitive Semantic Sensor Network ontology (CoSSN) ontology: CoSSN aims at overcoming the semantic interoperability challenges introduced by the variety of sensors potentially used. It also aims at describing expert knowledge related to a specific domain.
- Design and implementation of SeMoM: SeMoM is a flexible IoT architecture built on top of CoSSN ontology. It relies on a message oriented middleware (MoM) following the publish/subscribe paradigm for a loosely coupled communication between system components that can exchange semantic observation data in a flexible way.

From the applicative perspective, we focus on healthcare applications. Indeed, specific approaches and individual prototypes are preeminent solutions in healthcare which straighten the need of an interoperable solution especially for patients with multiple affections. With respect to these challenges, we elaborated two case studies 1) bedsore risk detection and 2) Activities of Daily Living (ADL) detection as follows:

- We developed extensions of CoSSN to describe each domain concepts and we developed specific applications through SeMoM implementing expert knowledge rules and assessments of bedsore and human activities.
- We implemented and evaluated the SeMoM framework in order to provide a proof of concept of our approach. Two experimentations have been realized for that target. The first is based on a deployment of a system targeting the detection of ADL activities in a real smart platform. The other one is based on ADLSim, a simulator of activities for ambient assisted living that can generate a massive amount of data related to the activities of a monitored person.

Résumé

De nos jours, l'internet des objets (IoT) connaît un intérêt considérable tant de la part du milieu universitaire que de l'industrie. Il a contribué à améliorer la qualité de vie, la croissance des entreprises et l'efficacité dans de multiples domaines. Cependant, l'hétérogénéité des objets qui peuvent être connectés dans de tels environnements, rend difficile leur interopérabilité. En outre, les observations produites par ces objets sont générées avec différents vocabulaires et formats de données. Cette hétérogénéité de technologies dans le monde IoT rend nécessaire l'adoption de solutions génériques à l'échelle mondiale. De plus, elle rend difficile le partage et la réutilisation des données dans d'autres buts que ceux pour lesquels elles ont été initialement mises en place. Dans cette thèse, nous abordons ces défis dans le contexte des applications de santé. Pour cela, nous proposons de transformer les données brutes issues de capteurs en connaissances et en informations en s'appuyant sur les ontologies. Ces connaissances vont être partagées entre les différents composants du système IoT.

En ce qui concerne les défis d'hétérogénéité et d'interopérabilité, notre contribution principale est une architecture IoT utilisant des ontologies pour permettre le déploiement d'applications IoT sémantiques. Cette approche permet de partager les observations des capteurs, la contextualisation des données et la réutilisation des connaissances et des informations traitées. Les contributions spécifiques comprennent :

- Conception d'une ontologie « Cognitive Semantic Sensor Network ontology (CoSSN) » : Cette ontologie vise à surmonter les défis d'interopérabilité sémantiques introduits par la variété des capteurs potentiellement utilisés. CoSSN permet aussi de modéliser la représentation des connaissances des experts.
- Conception et mise en œuvre de SeMoM : SeMoM est une architecture flexible pour l'IoT intégrant l'ontologie CoSSN. Elle s'appuie sur un middleware orienté message (MoM) pour offrir une solution à couplage faible entre les composants du système. Ceux-ci peuvent échanger des données d'observation sémantiques de manière flexible à l'aide du paradigme producteur/consommateur.

Du point de vue applicatif, nous sommes intéressés aux applications de santé. Dans ce domaine, les approches spécifiques et les prototypes individuels sont des solutions prédominantes ce qui rend difficile la collaboration entre différentes applications, en particulier dans un cas de patients multi-pathologies. En ce qui concerne ces défis, nous nous sommes intéressés à deux études de cas : 1) la détection du risque de développement des escarres chez les personnes âgées et 2) la détection des activités de la vie quotidienne (ADL) de personnes pour le suivi et l'assistance à domicile :

- Nous avons développé des extensions de CoSSN pour décrire chaque concept en lien avec les deux cas d'utilisation. Nous avons également développé des applications spécifiques grâce à SeMoM qui mettent en œuvre des règles de connaissances expertes permettant d'évaluer et de détecter les escarres et les activités.
- Nous avons mis en œuvre et évaluer le framework SeMoM en se basant sur deux expérimentations. La première basée sur le déploiement d'un système ciblant la détection des activités ADL dans un laboratoire d'expérimentation pour la santé (le Connected Health Lab). La seconde est basée sur le simulateur d'activités ADLSim développé par l'Université d'Oslo. Ce simulateur a été utilisé pour effectuer des tests de performances de notre solution en générant une quantité massive de données sur les activités d'une personne à domicile.

List of Publications

International Conferences

- Zgheib, Rita, et al. "A Flexible Architecture for Cognitive Sensing of Activities in Ambient Assisted Living." Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2017 IEEE 26th International Conference on. IEEE, 2017.

- Zgheib, Rita, Emmanuel Conchon, and Rémi Bastide. "Engineering IoT healthcare applications: towards a semantic data driven sustainable architecture." eHealth 360°. Springer International Publishing, 2017. 407-418.

- Faux, Francis, Rémi Bastide, Nathalie Souf, and Rita Zgheib. "Smartphone-Based Collaborative System for Wounds Tracking." 8th International Conference on e-Health, Telemedicine and Social Medicine (eTELEMED 2016), colocated with other events part of DigitalWorld 2016. 2016.

- Zgheib, Rita, Rémi Bastide, and Emmanuel Conchon. "A semantic web-of-things architecture for monitoring the risk of bedsores." Computational Science and Computational Intelligence (CSCI), 2015 International Conference on. IEEE, 2015.

Technical Reports

- Maxa, Jean-Aimé, Mohamed-Slim Ben Mahmoud, Nicolas Larrieu, Gilles Roudière, Rita Zgheib, Alinoé Abrassart, Bastien Mancini, and Sébastien Cayla. "Choix d'un protocole de routage pour le projet SUANETs (Security of UAV Ad hoc Networks)." (2014).

- Maxa, Jean-Aimé, Mohamed-Slim Ben Mahmoud, Nicolas Larrieu, Gilles Roudière, Rita Zgheib, Romain Laporte, Alinoé Abrassart, Bastien Mancini, and Sébastien Cayla. "Cahier des charges: Architecture de communication securisée pour une flotte de drone." (2014).

Conferences and Presentations

- Poster at Journées CPS Cyber-Physical Systems 2016, entitled « Semantic Data Driven Software Architecture for IoT Healthcare Applications » (Best Poster Prize).

- Oral communication at 9es Journées de l'Interopérabilité des Applications d'Entreprise – JIAE 2016. Presentation title : "Engineering IoT Healthcare Applications: Towards a Semantic Data Driven Sustainable Architecture" (Best PHD Student Presentation)
- Oral communication at 8es Journées de l'Interopérabilité des Applications d'Entreprise – JIAE 2015. Presentation title : "Message Oriented Middlware (MOM) sémantique pour la détection d'activité quotidienne (ADL)"

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

"The art and science of asking questions is the source of all knowledge."

— *Thomas Berger*

Contents

1.1 Background and Motivation

Over the past decade, the Internet of Things (IoT) paradigm [2] has continuously conquered the minds of researchers and engineers, to the point of becoming one of the most talked-about technology trends and exciting innovation in various fields as shown by the hype at recent International Consumer Electronics Show1 (CES) [1].

The European research cluster of IoT (IERC) defines the IoT as illustrated in Figure 1.1: “The Internet of Things allows people and things to be connected anytime,

[1] http://www.ces.tech/News/Press-Releases.aspx

anyplace, with anything and anyone, ideally using any path/network, and any service". In fact, in the Internet of Things [3], intelligent sensors are deployed on a smart space in order to gather real-time information that can be treated, recorded and correlated to infer new data that is more accurately processed and in a faster way. As the IoT expands and more devices (cars, kitchen, heart monitors, watches, etc.) across the globe become connected, it will have a particularly profound effect on world economy and businesses. Kranz points out in his book "Building the Internet of Things" [4] that IoT hype focuses on billions of inter-connected devices, driving digital transformation and trillions of dollars of economic opportunity. Although, the world's leading technology companies have found their business in IoT which is best exemplified in Google's Nest Learning Thermostat, in Watson IoT by IBM, in Apple Watch, and in home-shopping button from Amazon.

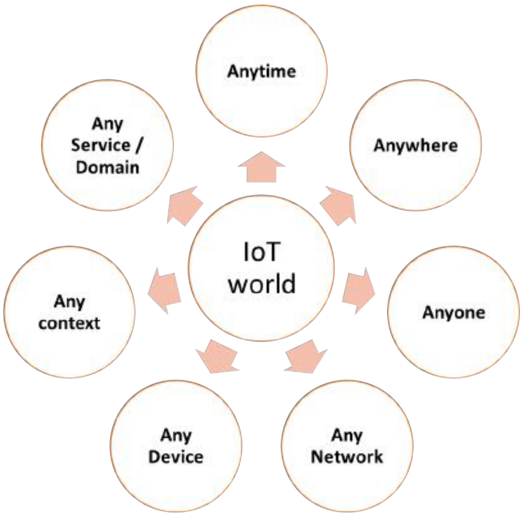


Figure 1.1 Definition of IoT world

IoT is gaining popularity and success in all domains, such as agriculture, defense, transportation, oil companies and healthcare. In particular, healthcare is among the fastest growing domains in the current time since it affects the whole world population and it cannot be fulfilled by applying traditional solutions anymore [5]. Innovative companies have also been involved in different e-Health applications and they are trying to capitalize on it. Namely, Home application and Brillo platform from Google; CareKit, Homekit and Apple Watch from Apple; and finally Amazon echo have found their interest in smart IoT products for healthcare. Recently Philips and Amazon have signed a collaboration project to innovate in e-Health, the aim of the project is to expand eHealth in the cloud [2].

Building IoT-based healthcare applications provides the possibility to improve people lives, from better personal safety, to being able to more closely monitor our health and that of our loved ones, to helping us save time and make better use of our natural resources. However, IoT in the healthcare domain presents several challenges related to the fact that almost every week a major vendor announces a new IoT strategy or division. The proliferation of ad-hoc and specific healthcare products, sensors and applications poses significant challenges pertaining to the complexity of designing and managing such systems, to the heterogeneity of the generated data, to the scalability, and the flexibility of the system to support the integration of highly distributed and heterogeneous data and knowledge sources.

The aim of this thesis is to provide an IoT architecture design that deals with these challenges and offer a flexible and interoperable communication system rather than an isolated one. It allows collaboration between IoT applications and facilitates the integration of novel applications that reason based on information from several data sources. This approach is applied to healthcare domain but it could be relevant to many other IoT domains.

In this introductory chapter, we present first IoT in healthcare context as the chosen application domain to illustrate the solution proposed in this thesis. Second, we state existing research and related works that aim to overcome the challenges related to designing an IoT architecture. Regarding the existing solutions, we introduce the main scientific challenges that we address and we present our position and the novelty of our proposal. Finally, we provide a road-map of the main scientific contributions of this work.

1.1.1 IoT in Healthcare context

The upcoming demographic change in the world has become a major concern and an important research field for modern societies [6]. In particular, the aging of population will impact social and economic balance of society due both to the increased burden on family caregivers and to the increase of healthcare costs [7]. As the global population ages, the demand for end-of-life care will continue increasing over the years. Elders desire staying in their own homes as long as it is possible and safe. To this end,

[2] http://ehealthnews.co.za/philips-amazon-ehealth-cloud-service/

steps must be taken to provide them with the necessary assistance, made possible by exploiting new technologies.

Moreover, billions of sensors including medical devices will be connected in the near future and will reach 50 billion by 2040 according to Garner statistics [8]. The increasing number of sensors that encompasses a large number of health measurements lead IoT to be an important driver of healthcare [9] [10].

Based on intelligent sensors, the combination of assisted living spaces and healthcare applications has been found to provide safe and quality care in the home. Within these applications Ambient Assisted Living (AAL), an IoT platform powered by artificial intelligence techniques enables monitoring health of elderly and incapacitated individuals in their place of living in a convenient and safe manner. Monitoring activities and behaviors of elderly in smart homes[11] [12] and fall detection [13] are successful examples of services provided by an AAL system. In addition, a significant proportion of elderly population suffers from age-related health issues such as Alzheimer's disease, bedsore, diabetes, cardiovascular disease, osteoarthritis, and different chronic diseases. The progressive decline in physical skills coupled with these diseases have led to innovate and create medical applications for disease monitoring at the hospital, retirement home or personal home.

A typical healthcare system based on IoT solution as presented in figure. 1.2 is characterized by deploying in the environment of monitored persons wearable and ambient sensors, each of them providing specific features and functionality required by any application. These devices are connected to the network through an IoT platform in order to communicate their data. An IoT platform helps integrating the IoT information in order to send real-time data to the applications for reasoning and making decisions. In the literature, designing IoT platforms present an active area of research since it presents several challenges due to the high heterogeneity of the system.

1.2 Overview of the Research

IoT world is a huge world that yields to more complex and heterogeneous systems of systems (SoS) [14] posing numerous challenges and issues that should be solved before having a full and complete solution ready to use. Until today, there is lack in presenting standards and unified architecture for IoT solutions, for instance in each application domain personalized and specific IoT applications have been developed.

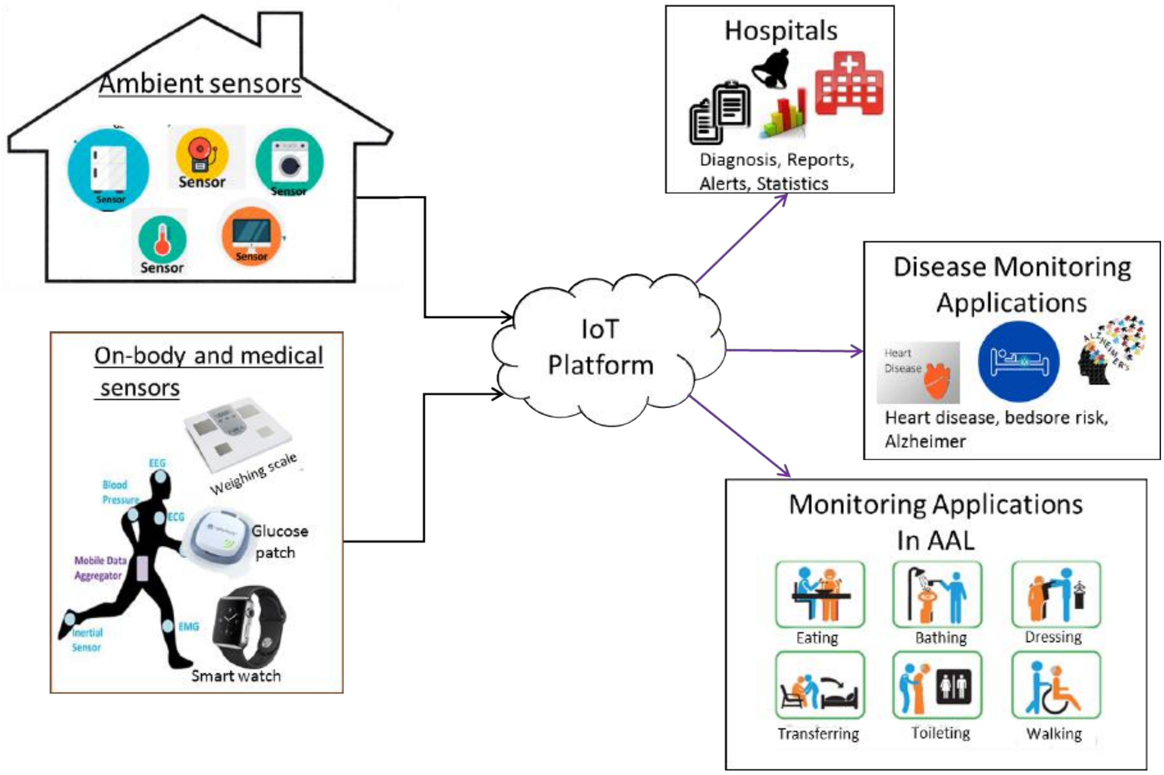


Figure 1.2 Healthcare system based on IoT technology

From an architectural perspective, middleware solutions have gained a reputation in the literature for providing credible solutions to manage the communication between IoT system components. They meet several requirements referred to system interoperability, scalability and availability of information in the system [15] [16] [17]. In the literature different middleware architecture have been proposed in that context and we will argue in Chapter 2 that the Message Oriented Middleware (MoM) with its publish-subscribe paradigm is the suitable solution that can offer a loosely coupled and performed communication between IoT system components.

Interoperability and scalability in IoT have been discussed from description perspective in some IoT projects. In the literature, different description techniques have been proposed to describe sensors metadata like XML schema and ontologies [18] [19]. Sensors description have been found to abstract the physical layer and provide a formal semantic description of physical sensors which enhance the interoperability over the IoT architecture. For example, the SSN ontology [20] has been designed to describe observations, stimulus and measurements of physical sensors. From other side, ontologies allow to describe domain-specific concepts such as transportation [21], cancer disease [22] and human activities [23]. Furthermore, sensor data can be presented in a formal way using semantic techniques such as XML, RDF, OWL and json formats. We have studied the different uses of semantic description and semantic techniques and we found that ontologies are the predominant technique for describing sensor data and metadata as well as describing domain description. Hence, an ontology-based solution can then bridge between the physical sensor deployment and the application specifications.

Nevertheless, IoT solutions still poorly consider semantic interoperability across IoT architecture which have been deployed independently from each other in the same environment. Moreover, as domain concepts and expert knowledge are pillar agents for an IoT healthcare system, adding knowledge description of these concepts would facilitate the integration of the domain concepts into the system.

1.3 Thesis Positioning

The recent advances in intelligent environment including sensing, communication and knowledge description have made the IoT a potential technology to enhance the quality of information in several domains including healthcare. However, challenges and requirements still need to be improved in order to achieve the expected performance of such environments. These challenges are related to the system interoperability and semantic interoperability of an IoT system, ensuring an optimal use of sensors. We state in this section the scientific challenges that will be discussed in this thesis as well as the scientific approaches that we consider to tackle these challenges.

1.3.1 Scientific Challenges

Heterogeneity of IoT components: Based on the study of the IoT world today and of the several proposed research approaches, we found that the adoption of the different available sensors technologies in the market is becoming more complicated and their integration in a same environment is even more difficult. Moreover, based on these specific, sensors ad-hoc IoT applications have arisen [24] making the integration and collaboration of the several IoT system components more difficult. The integration difficulty is related to the fact that each application is proposing a specific solution based on specific set of sensors, that means data is becoming proprietary for each application. Hence, sharing sensor data is becoming impossible which is not suitable for healthcare systems. For example, to monitor elders who often incur the risk of many pathologies symptoms and problems, a lot of sensors are required to provide the necessary data to the different diseases monitoring applications [25].

Lack in software engineering designs: As in other domains, the healthcare domain lacks in software engineering methods and good practices. In fact, in healthcare many IoT monitoring applications have been developed based on medical and ambient sensors that address a large number of human health measurements. Such application focuses on data processing and clinical validation, and it does not address necessarily the way data is gathered.

Definition of flexible IoT architecture: Due to the heterogeneity nature of the various IoT components (sensors and applications), defining an interoperable and **flexible IoT architecture** represent an open research challenge. Especially, the integration of specific domain concepts such as healthcare is a problematic when designing an IoT architecture. This issue is related to the need of extracting a meaningful knowledge and detecting the health condition of the monitored person when data from different sources is needed and each one is presented in a specific format. Moreover in a domain like healthcare expert knowledge is essential and in most cases a collaboration between IoT programmers and medical staff is needed to add medical knowledge to the system which make complicated the development and implementation tasks. The challenges related to this context are summarized as follows:

- The interoperability of the whole system;
- The scalability issue and supporting the huge number of sensors ready to connect;
- The flexibility and the easy integration of new devices;
- The availability of the information and the manageability of resources ensuring the safe transmission of data to the corresponding data consumers.

1.3.2 Scientific Approaches

Loosely coupled communication: In order to prevent ad-hoc application design with specific solution, our objective is to define a flexible IoT architecture based on sound software engineering practices and to comply with the principles of weak coupling between software components. In that way, each sensor can be used by many applications and the information can be provided by different data sources. This approach enhances the sharing of information between an IoT system components, reduces redundant sensors and provides a bridge between physical world, real applications and experts knowledge.

Data and Domain description: To enhance the semantic interoperability through IoT architecture, we advocate a semantic description of sensor data and meta data. In that way, all system components exchange formal messages based on observations of physical sensors. Moreover, our description method rely on ontology technique and allow to describe domain concepts and expert knowledge in a domain like healthcare. Regarding the healthcare domain, our objective is to facilitate the integration of IoT and intelligent system in the medical process such as monitoring diseases and healthcare factors. Moreover, our objective involves a better management of public health economy by following prevention methodology which is cheaper then treatment costs usually.

1.4 Thesis Scientific Contribution

The major contributions of the thesis are presented as follows:

- Chapter 2: We present a state of art of the various domains and perspectives that serve as a background of our research, healthcare applications, middleware architectures and semantic description.

  We present the healthcare domain and the IoT system requirements in this domain. We aim to provide a better understanding of the role of IoT and intelligent systems in improving healthcare by providing the ability for elderly to "aging in place" thanks to the alert systems and monitoring applications needed to follow up the monitored person. In addition, we show the lack of software engineering methods and IoT architecture design in this domain.

  We review a state of the art of the different middleware solutions proposed in the last decade for IoT systems. We state the major requirements and design issues for middleware support in the intelligent environments that have been extensively discussed in the literature. We make a comparison between SOA and MoM the most proposed middleware for IoT.

  We provide a summary on the different contexts of semantic description in IoT systems: I)Sensors modeling and descriptions II)Exchanged messages description and III) Domain description. We show that the three description perspectives should be considered in a smart IoT ontology to improve the smartness of the system by unifying the format of data sharing and describing different levels of knowledge.

- Chapter 3: We define a sensor hierarchy illustrating the several possible "things" that can be connected in IoT. we define and develop the Cognitive Semantic Sensor Network ontology (CoSSN) ontology, aiming at overcoming the semantic interoperability challenges introduced by the variety of sensors potentially used and at representing expert knowledge and the cognitive process to process any domain factors and assessments as actionable knowledge. We illustrate our semantic model by a medical application use case with the aim to detect the bedsore development risk for the elderly.

- Chapter 4: We define SeMoM, a Semantic Message Oriented Middleware architecture for IoT applications. This architecture is well suited for IoT applications and tested in the context of the healthcare domain. The architecture relies on a message-oriented middleware following the publish-subscribe paradigm for a loosely coupled communication between system components that can interchange semantic observation data in a flexible manner. Hence, this architecture rely on the defined CoSSN ontology providing the mechanisms to describe semantically sensor data and metadata as well as domain concepts. MoM allows the integration of sensors and sensor data in such a way that the same sensor can potentially be used for several applications, thus reducing the number of connected sensors. CoSSN allows the interoperability of sensor data to ensure that the same information can potentially be provided by different sensors ensuring the redundancy of data in the case of an out-of-service sensor. Furthermore the proposed approach allows to define cognitive sensor components that are able to receive semantic data, apply rules through decision modules and infer higher-level semantic knowledge, such as human activities and bedsore risk.

- Chapter 5: We design and develop a platform regarding the principles of SeMoM architecture. The target of this platform is to provide a proof of concept of the proposed solution. Another important aspect of this platform is its capacity of abstracting the programming and knowledge process. The platform provides means to the programmer to develop and deploy a semantic IoT system in a flexible and easy way. We present also the activity and behavior detection case study in AAL context. It is used for experimentation and evaluation purposes in this thesis.

- chapter 6: We conclude our dissertation by summarizing all the work achieved during this thesis and we present the perspectives and future work that will follow this research work.

Chapter 2

Background and Related Work

"There is no science which does not spring from pre-existing knowledge, and no certain and definite idea which has not derived its origin from the senses.."

— *William Harvey*

Contents

2.1 Introduction

Physical sensors and applications are the main two actors in an IoT system. Exchanging data between these two components make IoT paradigm an intelligent environment where applications are able to generate high level knowledge from raw sensor data. The main recognized challenges in this environment refer to the heterogeneity of technologies and devices that are present in an IoT network. This heterogeneity makes the management of sensors and the communication between them very complex. Therefore, our aim is to handle these challenges by providing an interoperable IoT architecture offering loosely coupled communication enhanced with semantic techniques. This combination would enhances the interoperability and manageability of an IoT system.

Research into the IoT is still in its early stage, and no standard definition of the IoT is widely agreed upon. However, we can find a lot of attempts and propositions that tackle different aspects of IoT management challenges. In this chapter we explore the different fields that serve as a background for our work and the various approaches that have been proposed in the last decades related to the challenges that we are interested in. These challenges refer to technical and semantic interoperability as well as to sensors integration for IoT monitoring systems [26] that we have stated in chapter 1. For that, we first introduce in section 2.2 IoT systems in the healthcare domain, with its main components: sensors, gateway and applications. We provide in section 2.3 a review on the existing middleware approaches for IoT systems and in the healthcare domain. We then analyze in section 2.4 several techniques for sensors and data description and the different perspectives of semantic description in an IoT system. We state the ongoing challenges and issues in section 2.2. Finally, we conclude this chapter by identifying the existing limitations and our position regarding the presented state of the art in section 2.6.

2.2 IoT system in Healthcare

Haller et al. [27] defined the term Internet of things (IoT) as “A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business processes”. Singh et al. [28] referenced the IoT term as a network of physical sensors or objects called things (e.g., smartphones, bracelets, TVs, cars). These things are able to communicate and interact with the real world in order to generate data to applications.

IoT is a general term and can target different domains. In the context of healthcare, an IoT system often consists in the deployment of specific physical sensors that encompass health measurements and patient information. The system can de deployed in the environment of the patient, in a hospital, at home or in a retirement home. For instance, IoT applications connected to the network are mainly targeting reasoning on health measurements by applying assessments and algorithms for detection of diseases or monitoring a patient status.

Our objective is to design an IoT architecture for healthcare applications that improves the interoperability and the sharing of information between all components. To achieve this goal, it is necessary to define the various actors that can interact in an IoT system. In every application domain, it is possible to distinguish three main actors in IoT whatever the domain is, as shown in Figure 2.1. These actors refer to **physical sensors**, communication component or **gateway** and **IoT applications** that request sensor data.

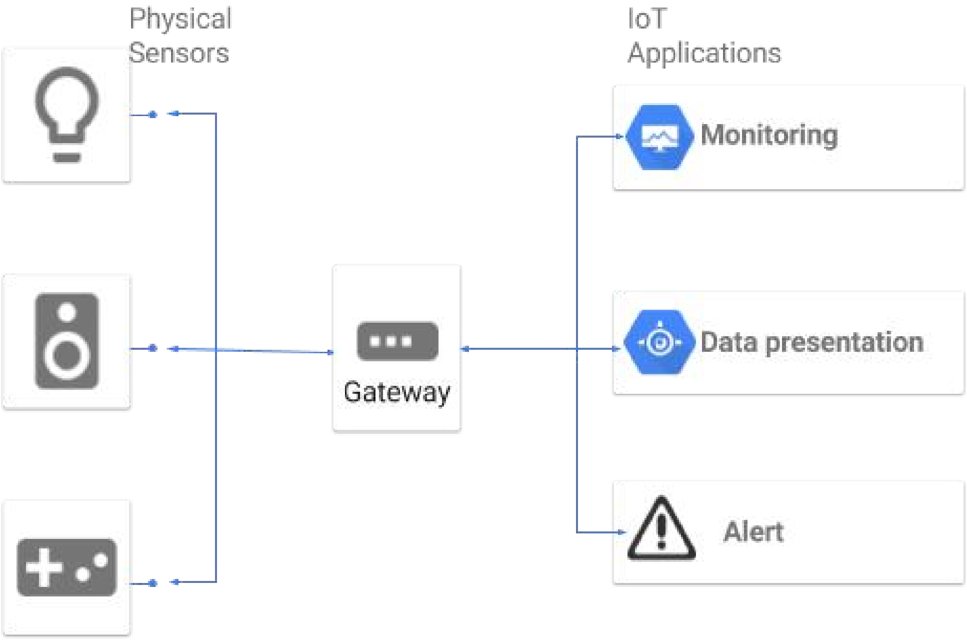


Figure 2.1 Overview of an IoT system

2.2.1 Sensor Networks

A sensor is a smart object that can communicate, sense and interact with the external environment by means of suitable information and communication technologies, to enable a range of applications and services. Garner [8] forecasts that the IoT will reach 50 billion units by 2040, therefore IoT sensors will be an important part of the future solutions and strategies in every application domain. For example, a pedometer, a connected bracelet or a smart watch can be used to monitor the physical activities, fitness and heart rate of a person. In this context, we can already find off-the-shelf several sensors proposed by Fitbit, Jawbone Up, Withings, V-Patch and others.

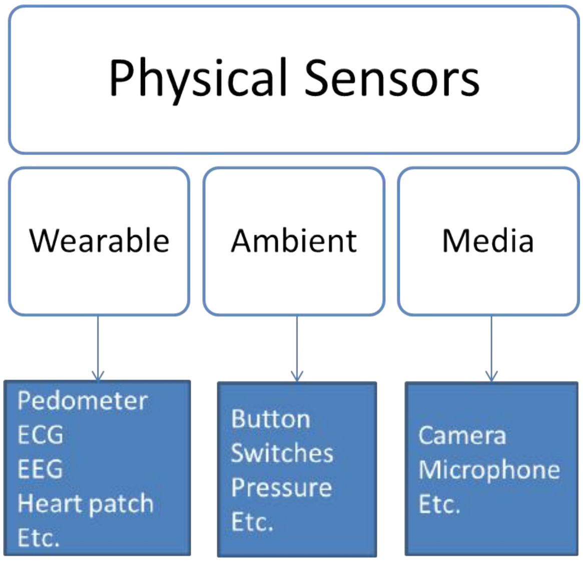
Physical sensor networks are composed of spatially distributed sensors that can be deployed in an intelligent space environment to monitor environmental or physical conditions of a person. It is the main source for data acquisition in healthcare systems. Physical sensors can be classified into media, wearable and ambient sensors.

The first category is **media**-based approach which is one of the most used technique in monitoring systems. It basically relies on visual surveillance, video and sound retrieval, smart TV or human-computer interaction techniques [29]. This approach has gained success in some projects [30] [31] [32] and presents a relevant technique with remarkable features for monitoring systems especially for activity recognition. But, it is usually not considered acceptable for public. Deploying cameras and sound sensors often instills a disruption and uncomfortable feeling for elderly people [33]. The aforementioned privacy concerns have led researchers to come up with new solutions based on non-intrusive sensors.

The second and third category are more acceptable and many approaches rely on it. With solutions based on **wearable devices** [34], [35], sensors are usually present in bracelets, wrings and watches but can also been found in textile like a coat or a T-shirt. A person should wear these objects in order to get measurements related to its movement and physiological signs. In some studies, wearable devices can be referenced as body sensors and body area networks. There is a number of survey publications [36] [37] that have reviewed the work done so far in this area. Initially, dedicated wearable sensors were used to recognize human physical activities, heart rate and sleeping. However, there has been a shift towards mobile phones in recent years. The availability of various sensors in these devices such as GPS, accelerometer, gyroscope and pedometers, has encouraged the development of smartphone based applications that aim to detect person's activities. This has been achieved successfully for sport activities [37] like running and walking using pedometers for step counting. A survey on these solutions can be found in[38].

Ambient or ubiquitous sensors are increasingly prevalent in monitoring environments due to the low cost of these sensors and their ease of installation[39]. Based on these sensors, we can detect different states like "door is open" or "oven is on", person's position and environmental attributes like temperature, light or humidity. In the context of activity recognition [40] [41], motion, force, vibration, water, buttons and other sensors provide information about an individual state and actions. Information gathered from each sensor is then merged and correlated in order to recognize the activity of daily living of the person, for example, meal preparation, sleeping, watching tv, taking a snap, eating and so on.

Recent solutions try to mix the advantages of different kinds of sensors. For instance, Shoaib and al. in [42] propose to process human activity recognition using smartphone inertial sensors and wrist sensors. Likewise, authors in [43] propose to use wrist sensors like smart watches and video based cameras for the same purpose.


Physical Sensors

Wearable

Ambient

Media

Pedometer
ECG
EEG
Heart patch
Etc.

Button
Switches
Pressure
Etc.

Camera
Microphone
Etc.

Figure 2.2 Classification of the possible deployed sensors in an IoT healthcare system

Physical sensors presented in Figure 2.2 belonging to any before mentioned category present a pillar component in an IoT system. This fact has encouraged many vendors to propose new types of sensors with new technologies and infrastructures each day.

Therefore, several communication technologies and protocols have been proposed to enable sharing of sensor data.

2.2.2 Communication Technology Heterogeneity

Physical sensors deployed in IoT environment must communicate data to healthcare applications in order to perform analysis. Two types of communication can be found between sensors and applications: direct communication and indirect communication.

Ad-hoc prototypes are common in healthcare because most researchers are mainly concerned with proving that some medical assessments can be automated with IoT. In such cases, biosensors are deployed and the application is developed with a **direct communication** with sensors. Therefore, there is little consideration of good practices or architecture design in such applications. The study in [44], is an example of these approaches where authors were making tests on RFID sensors and their usefulness in the healthcare domain. Moreover, these applications usually rely on one kind of sensors for a simple management system.

Indirect communication through a gateway between sensors and applications has been widely discussed in the literature. Nobuo Saito in [45] reviewed the home gateway from a broad and practical perspective. A gateway is responsible for network interconnection, network management and application management. Its main functionality is the coordination between heterogeneous sensors and connecting them to the network. Several wireless communication technologies are available to serve data transmissions among sensors and gateways in personal area and body area networks in healthcare network. The most popular and used technologies are the short-range wireless protocols such as Zigbee, Bluetooth, WiFi [46] [47], and recently Bluetooth Low Energy [48] and LoRa [49]. The characteristics of these technologies are presented in Table 2.1.

Other wireless technologies are selected for specific applications such as the identification and tracking of persons and objects using RFID, IrDA, and UWB. These technologies are used together with large-scale wireless networks such as 3G/4G to provide advanced and pervasive healthcare applications and services. Figure 2.3 presents an overview of the different protocols and standards for communication used in IoT network, based on the OSI model.

Table 2.1 Characteristics of some wireless technologies.

Technology	**Zigbee**	**Bluetooth**	**BLE**	**WiFi**	**LoRa**
Standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.15.1	IEEE 802.11	IEEE 802.15.4g
Frequency Band	868/915 MHz/2.4 GHz	2.4 GHz	2.4 Ghz	2.4/5 GHz	868/915 MHz
Bit rate	20/40/250 kbps	1-3 Mbps	1 Mbps	11/54 Mbps	0.3-50 kbps
Network Topology	P2P, star, mesh (tree)	Piconet	Piconet	P2P, star	P2P, star
Maximum nodes	>64000	2-8	Implementation dependent	32	Implementation dependent
IP enabled	NO	NO	NO	YES	YES
Optimized for	Low power and low cost, relia- bility, and scalability	Convenience	Low cost, low power	Speed, flexibility	Low power

IoT world is witnessing a proliferation of physical sensors with different telecommunication technologies. One IoT network may include a huge number of physical sensors from different vendors and communicate in a specific communication technology. Hence, as a gateway is in charge of managing an IoT network it should be designed in such a way to accommodate the various technologies presented in the IoT market. Moreover, healthcare applications connect to the gateway with the aim of gathering the needed information from sensors. An IoT architecture should be also able to handle various potential healthcare applications that can be connected to the gateway. The number and the heterogeneity of the components in IoT is still challenging until today and most of solutions are unable to deal with this diversity and heterogeneity.

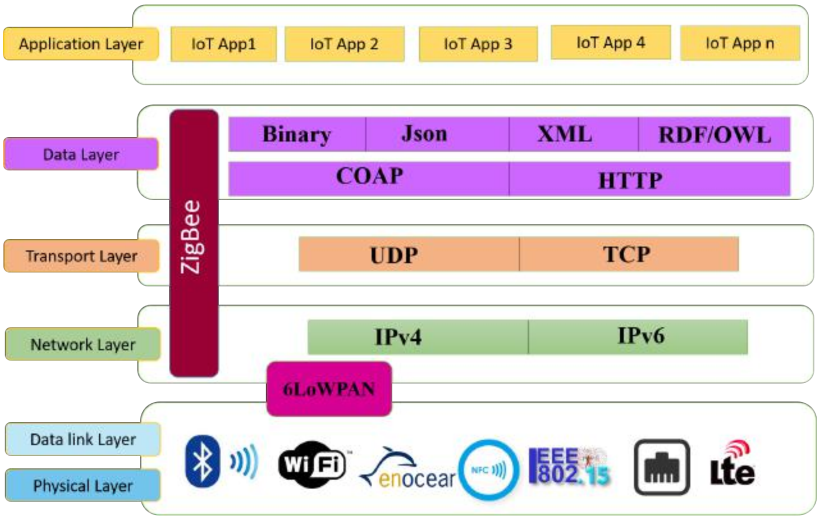


Figure 2.3 Overview on Standards and communication protocols among IoT systems

2.2.3 Healthcare Applications and Processing Systems

Powered by IoT's identification, sensing, and communication features, all subjects in healthcare systems such as people, equipment and medicine can be monitored [50]. By using the personal computing devices (laptop, mobile phone, tablet, etc.) and mobile internet access (WiFi, 3G, LTE, etc.), the IoT-based healthcare services can be mobile and personalized [51]. Thanks to these advanced technologies, several IoT applications have been developed in healthcare context. The aim of these applications is to collect, manage, and share efficiently all the healthcare-related information (logistics, diagnosis, therapy, recovery, medication, management, finance, and even daily activity) [52]. For example, a patient's heart rate can be periodically collected by sensors and then sent to the doctor's office.

Healthcare applications can be classified into **Medical applications** and **Ambient Assisted Living (AAL) applications**. Medical applications are mainly installed at hospitals and retirement homes to prevent, detect diseases or even monitor patients health condition. AAL applications are mainly installed at patient home to monitor and follow his/her health and activity at home. Healthcare applications gather sensor data in order to apply some **reasoning techniques** to generate diagnosis.

- **Medical Applications:** Health monitoring devices range from blood pressure and heart rate monitors to advance devices capable of monitoring specialized implants such as pacemakers, fit bit electronic, wristbands or advanced hearing aids. Whereas medical application through monitoring diseases helps improving healthcare by alleviating nurses job at retirement home and hospitals which enhances the economical management in heath institutes, it can be witnessed also at home.

  For health applications, lots of solutions [53] [54] have already been proposed targeting asthma, Alzheimer, cardiology, bedsore, fall detection and ADL detection. As an example, LOBIN [55] is a novel healthcare IoT platform which allows monitoring several physiological parameters, such as ECG, heart rate, body temperature, etc. It consists of a set of devices (namely distribution points (DPs)) which transmit data via a gateway that forwards them to the management subsystem. Other examples such as UniversAAL [56], REACTION [1] and SOPRANO [57] also present IoT health applications that we detail in chapter 4.

[1] http://www.reactionproject.eu/news.php

An illustration of a medical application is presented in Figure 2.4. In this example, on-body (bracelet, patch) and ambient (weigh) sensors are deployed in the environment of a diabetic patient to detect health factors such as blood pressure, blood glucose, body weigh and pulse rate [58]. Based on reasoning techniques, these factors are analyzed and correlated to create a personalized profile to assess behavioral and social aspects relevant to diabetes care.

In this thesis, we rely on bedsore risk detection as a potential medical application and an illustrative case study. The case study will be more detailed in chapter 3.

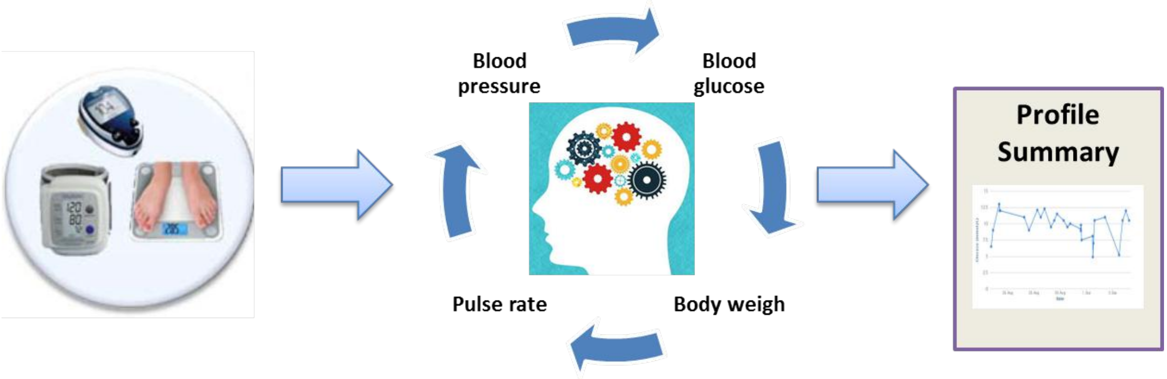


Figure 2.4 Monitoring of a diabetic patient based on connected devices

- **Ambient Assisted Living Applications:** Based on intelligent sensors, Ambient Assisted Living or AAL applications have been found to enhance safety and quality of care in the home. In [59] authors show that smart objects and IoT are an essential factor for AAL to support elderly people in their daily routine and to allow an independent and safe lifestyle as long as possible. In this context, monitoring applications such human activity detection have been developed with the aim to monitor elderly and follow their health and activity status at home. This approach has an important effect on the pubic health economy since it avoids unnecessary visits to hospital [60]. Moreover, it encourages prevention approach rather than curing diseases.

  In the literature, several projects have been introduced and tested for AAL. For example, authors in [61] propose an AAL system to train elderly people to handle modern interfaces for Assisted Living and evaluate the usability and suitability of these interfaces in specific situations, e.g., emergency cases. PERSONA [62] is another example which aims at monitoring the activities of person at home. More details can be found in the following surveys [7] [63].

Android-based and mobile-based applications such as [64] [53] [65] have also been addressed in AAL projects. The aim of this approach is to send sensor data to applications running on a smart phone for further processing and visualization. A target application of that work is the efficient monitoring and management of chronic diseases at home.

In the context of home care applications, one of the applicative objective of this thesis is long-term monitoring of elders by detecting and monitoring their Activities of Daily Living (ADL) using ambient sensors. This application will be detailed in Chapter 5.

2.2.3.1 Reasoning techniques in IoT applications

Data collection and integration are the main key technologies and an important stage in an IoT system [66]. In fact in IoT, sensor data is collected and processed using reasoning techniques in order to infer new actionable knowledge or information needed to predict an event or to describe another one. These techniques need to be developed to make sense of the collected data. For example, when a temperature sensor sends 45 °C as a value, a reasoner treats this information by applying rules and inferring new useful information such as "temperature is high and palliative steps must be taken". Processing information and reasoning about data originating from multiple sensors can provide higher accuracy than a single sensor, and is mandatory in order to detect complex pathologies and activities of monitored person. In the literature, many reasoning techniques have been proposed. Machine Learning and semantic techniques are leading solutions at the time of writing.

- Machine-learning techniques: Most of existing solutions rely on machine-learning techniques [67] for data processing. It has been applied to learning associations between factors and correlations of relevant contexts. These methods span a broad range of techniques from simple algorithms to more advanced methods like hidden Markov models (HMMs) [68] with several of its extensions, Bayesian networks [69], decision trees, neural networks and so on. The most distinguishable features of machine learning techniques is their ability to correlate and combine sensor data and situation and extract categorical activities in AAL context for instance such as running or walking. However, these techniques usually require a long period of learning before figuring out the rules that predict the activities.

Moreover, it requires big amount of examples stored as dataset in order to have good predicting model which is difficult to obtain.

- Semantic techniques: An important aspect of existing AAL proposals is the use of semantics and knowledge-driven models for domain concepts description. Ontologies in this strand have been widely discussed and many ontologies-based solutions have been proposed [70], [71], [72]. With respect to other solutions, knowledge-based methods are semantically clear in modeling and representation as well as highly effective in inference and reasoning. They create a complete and accurate model of an IoT application [73] that enhances the semantic interoperability of an IoT system whitout a prior learning phase. Semantic reasoning can be expressed as queries using the SPARQL query language [74]. It also allows to define domain rules such as defining the process of activity detection or the bedsore assessment. SWRL is another semantic technique that allows defining rules and applying reasoning to sensor data described by ontology means.

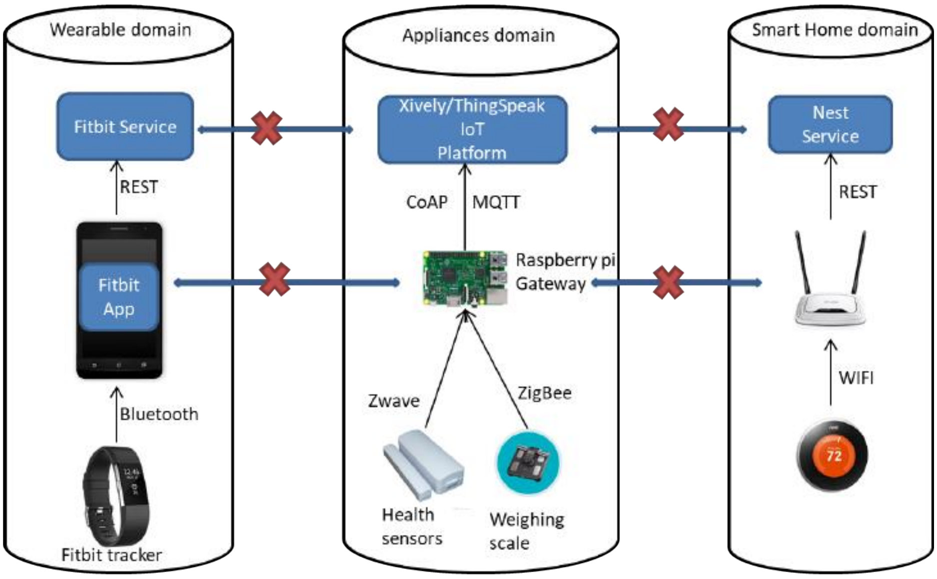


Figure 2.5 Heterogeneity issue in IoT Healthcare systems

Discussion

Many different sensors from different vendors have been proposed in the market and can be connected in an IoT healthcare system. Each one of them produces different values with different accuracy, precision and units of measurement [75]. Moreover, various communication technologies and standards are used to communicate sensor data to the network. As we have illustrated in Figure. 2.5 applications from different domains are not able to inter communicate and share their information. This problem is related to the complexity of managing set of components [76] and the heterogeneity of data sources or sensors. From the other side, several monitoring applications can be connected to a gateway searching for one or more sensor data. Hence, cross accessing to sensors information and management of redundant sensors in such environments present real challenges to design a gateway. This fact has strengthened the necessity of software architecture solutions offering flexible communication and high performance management of sensors [77]. In the literature, middleware solutions have been proposed to address these issues.

2.3 Middleware Architectures for IoT systems

Middleware [15] solutions provide a technical infrastructure that mediates between two or more systems. Their historical role is to ensure the transport of a message from one subsystem to another, with a more or less important level of coupling. Researchers find middleware as a suitable solution to fill the gap of heterogeneity and manageability of sensors because it provides an abstract layer interposed between the network layer and the application layer. Moreover, It aims to hide the technological details of communication technologies to enable the application developers to focus on the development of the IoT applications. As our aim is to provide an IoT architecture, it was necessary to state the main challenges to design an IoT middleware as well as the existing middleware approaches in IoT world.

2.3.1 Middleware Challenges

Middlewares are an interesting solution for IoT communication between distributed system components thanks to their ability to comply with several requirements in such environments. Several middleware architecture have been proposed in the literature, each one address some of the discussed challenges in an IoT environment such in [78]. They are regarded as essential problems to be solved before presenting an approach as a final solution.

Indeed, these challenges refer to the characteristics of IoT infrastructure and IoT applications:

1. Infrastructure Challenges:
   - Interoperability: The interoperability challenge is important for IoT middleware since many heterogeneous devices will communicate and exchange information together.
   - Scalability: Since a large number of devices are expected to be supported by IoT application, a reliable middleware is required to manage these devices and the addition of new ones.
   - Spontaneous interactions: In IoT applications, objects and devices can have spontaneous interactions and events. For instance, a smartphone user can come in close contact with a TV/fridge/washing machine at home that can generate events without the user's involvement. An IoT middleware should be capable of proceeding and acting at time after such events.
   - Unfixed infrastructure: The existence of many devices that have a specific infrastructure and many of which will be mobile, wirelessly connected, and resource constrained gives the dynamic nature to an IoT network. Mobile nodes within the network leave or join anytime they want. IoT middleware would be designed in a flexible way to handle the dynamic characteristic of the network.
   - Abstraction provision: An ideal Middleware for intelligent environment should give abstractions at many levels such at heterogeneous devices, interfaces, data stream, physicality and development process.
2. Applications Challenges:
   - Availability or Multiplicity: To guarantee the availability of services and information all the time, an IoT middleware must ensure the multiplicity of services and data sources.
   - Reliability: An IoT middleware should remain operational even in the presence of failures. It is necessary to have validate each component of the middleware in order to achieve high level system reliability including communication, data, technologies and devices from all layers.
   - Real-time: For many IoT applications (such as transportation or healthcare) real-time delivery of information is critical. Delayed information in such applications can be useless and sometimes dangerous. For example, a delayed notification of fall monitoring application can lead to the death of the person.

- Security and Privacy: Many devices are communicating with each other; lot of information is shared between them. This information can be private, personal and specially information about daily life. So trust, security and privacy are primordial challenges that should be taken into consideration in IoT middleware solution.

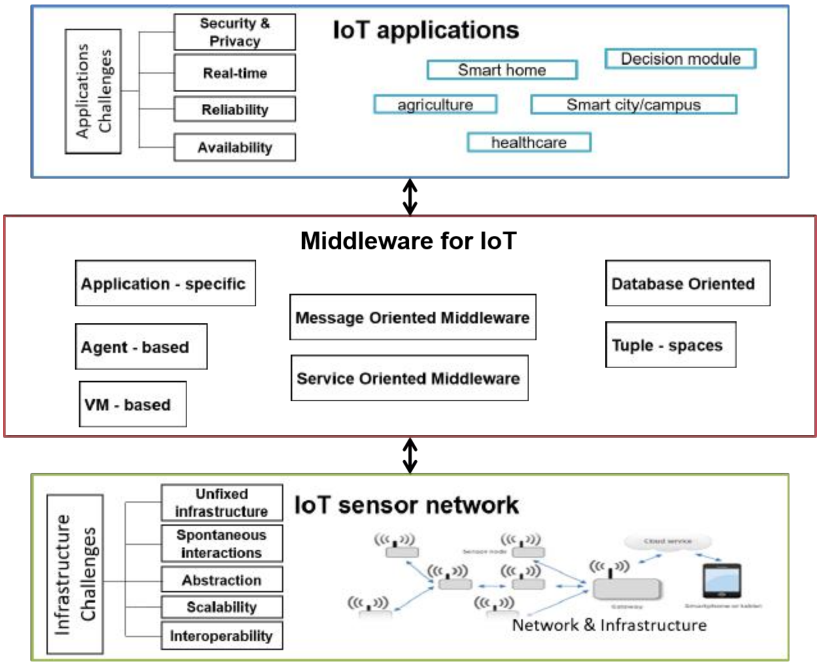


Figure 2.6 Relationship between IoT Data sources and IoT applications via the middleware mediator

In light of the presented challenges, in this thesis we are studying and providing solution for some of them. From the infrastructure perspective, we mainly address the interoperability and scalability issues. From the application perspective, we address the availability and real-time challenges.

2.3.2 Overview of middleware approaches for IoT

With respect to the aforementioned challenges, we have studied existing projects aiming to develop middleware solutions for IoT. As a result, a state of art has been elaborated in order to identify the different technologies and architectures proposed in this field.

Among a number of studies, a specific one is presented by Razzaque et al. in [79]. Authors in this paper have analyzed in details several existing solutions and have presented a recent overview of the different types of middleware architecture.

Another classification of middleware approaches in IoT is presented in [80] and [81]. Thanks to these studies, we have identified the different types of middleware (see Figure 2.6) for intelligent environments that can be classified as follows: Application-specific, Agent-based, VM-based, tuple-spaces, database-oriented, Service-oriented architecture and Message-oriented Middleware.

2.3.2.1 Middleware Architectures for IoT

Application-specific: This approach is based on the needs of a specific application and focuses on resource management. Thus a strong coupling exists between application and data providers which leads to specialized middleware.

MidFusion [82] represents an example of this middleware approach. It discovers and selects the best set of sensors or sensor agents on behalf of applications. It provides a QoS support only for networks based on Bayesian algorithms. Other middleware in the same category can be found in [83] which focuses on applications in smart home. It uses application-specific functions to choose a specific context. Given multiple alternatives, one alternative at any time that provides the context for all applications, whilst maximizing the applications' total "satisfaction" with the quality of context from the chosen provider. In [84], authors propose MiLAN, a middleware based on tight coupling. It allows applications to specify a policy for managing the network and sensors. They argued that the needs of the application should be integrated with the management of the network into a single unified middleware system.
This kind of architecture does not satisfy IoT middleware requirements due to its tight coupling nature between applications and data providers as we have shown in section 1.2.2. Especially it does not address the heterogeneity characteristic of IoT environment.

Agent-based: Agent or modular based approach [85] consists in applications division into modular programs to promote distribution and injection through the network using mobile agents. Impala [86] , Smarty messages [87] and Agilla [88] are examples of this approach. Many advantages towards IoT systems have been recognized of this approach, they can be highlighted by providing decentralized systems capable of tackling availability, reliability, and resource management requirements of middleware. This approach can reduce the complexity of a middleware architecture. However, it presents some limitations related to its inability to perform code management tasks and the unpredictability of agents in the system at runtime.

VM-based: This approach is flexible and contains virtual machines (VMs), interpreters, and mobile agents. Based on this approach, the middleware is composed of two layers. Each physical device is deployed as a VM in the first layer of the middleware. In the second layer, a general VM interprets the modules and delivers data to the application that expresses its needs by a query. This approach addresses architectural requirements such as high level programming abstractions, self-management, and adaptivity while supporting transparency in distributed heterogeneous IoT infrastructures [89] [90]. However, this approach suffers from the overhead that the exchanged instructions introduce.

Tuple-spaces: A tuple space [91] is a data repository that can be accessed concurrently. In the context of middleware, each device from the physical layer is represented as a tuple space in the middleware. All the tuple spaces form a federated tuple space on the gateway. This approach suits mobile devices in an IoT infrastructure, as they can transiently share data within gateway connectivity constraints. TinyLime [92] and TeenyLIME [93] are tuple-space middleware solutions for mobile ad hoc networks and sensor networks. Although they have flexible architecture that allows middleware to be used in different environments, they address frequent disconnections and asynchronous communications problems. However, they offer limitations in resource management, scalability, security and privacy.

database-oriented: This middleware approach considers the whole sensor network as a distributed and virtual database. It uses SQL like queries to collect target data over the network. GSN [94] is a database-oriented middlware that has been integrated in other projects such as OpenIoT [19]. While this approach offers good programming abstraction and good data management support, the remaining IoT requirements are not necessary addressed like scalability, real-time and spontaneous interactions. Moreover, its centralized nature makes it difficult to handle dynamic and heterogeneity characteristics of the IoT network.

The before mentioned architectures witness have been used in many specific IoT projects and research area. However, the important use of Service Oriented Architecture (SOA) and Message Oriented Middleware (MoM) solutions is remarkable in IoT projects in the past decade. A comparison between the two approaches is presented in the following section.

2.3.2.2 SOA vs MOM Middleware Architectures for IoT

Service-oriented Architecture: Two major trends in the world of IoT have been witnessed in the past years [95], first hardware is becoming smaller, cheaper and more powerful, and second the software industry is moving towards service-oriented integration technologies. Service-Oriented Architecture (SOA) is a way of thinking and designing the Information System and has been traditionally used in corporate IT systems. The key concept of SOA is the service which is a distributed software invocable and sited. In IoT approaches based on SOA, intelligent sensors are depicted as services for consumer applications. What is key to these services is their encapsulated nature, i.e. the service interface is independent of the implementation. Providers describe their services (sensors characteristics) and put them on service to the consumers. Web Services Description Language (WSDL) is the standard used to describe the services [96].

A state of art of SOA-based middlewares for wireless sensor network have been stated and described in [97], it refers to *SStreaMWare*, *USEME*, *SensorWeb 2.0*, *OASiS*, *B-VIS*, *MiSense*, *SOMDM (SI)*, *SOA-MM* as middleware proposals. These solutions have been proposed for wireless sensor networks. Their main features are supporting real-time monitoring, management of heterogeneous devices, data collection and filtering. However, the survey demonstrates that none of these solutions covers all the requirements of management of sensors network in intelligent environments.

More recent approaches have tried to enhance the SOA features and adapt it to IoT. For example, SenseWrap [98] provides a standardized communication interface to hide the sensor-specific details from the applications. It introduces the concept of virtual sensor to offer transparent discovery of sensors. However, virtualization is applied only to sensors, not to actuators or computing resources which makes it not fully suitable for IoT environments. TinySOA [99] employs simple and deterministic mechanisms for WSN resource (e.g., sensor nodes) registration and discovery. It supports only a few basic functional requirements (e.g., abstraction, resource discovery, and management). SensorsMW [100] has been found as an adaptable and flexible middleware for sensors management. It allows easy and efficient configuration of wireless sensor networks for information gathering. However, the reconfiguration may fail in critical applications as they define strict QoS rules. MOSDEN [101] supports a sensing as a service model built on top of GSN. It is based on a plugin architecture which improves the scalability and user friendliness. However the predefined resource/service discovery and service composition mechanisms features presented in this approach may be challenging in a dynamic IoT environment.

In healthcare, specific platforms based on middlewares have been developed. As an example, the *Sphere* project [102] implements a specific platform that follows a clustered-sensor approach. It aims to build a generic platform that fuses complementary sensor data to generate rich datasets that support the detection and management of various health conditions.

Amigo middleware [103] is proposed for smart home application and SM4ALL middlware [104]. SM4ALL has been developed and build on the OSGi/UPnP standards for building smart homes in the context of European Framework 7 project and Smart Homes for All (SM4ALL) project. It targets Ambient Assisted Living (AAL) environments to monitor disabled elders. LinkSmart [105] within the REACTION european project is another SOA-based middleware with the aim of monitoring and managing the diabetes of patients as well as their therapy in operational healthcare environments. It has been also used recently for a smart home application [106].

VIRTUS [107] is an event-driven middleware based on the standard XMPP (eXtensible Messaging and Presence Protocol). It guarantees a (near) real-time, secure and reliable communication channel among heterogeneous devices. In [9] [108], the authors present a platform based on cloud computing for management of mobile and wearable healthcare sensors, demonstrating this way the IoT paradigm can be applied on pervasive healthcare.

In [109], a SOA based solution offering a distributed telemonitoring system that aims at improving healthcare and assistance for dependent people in their homes. Another example [110] presents a pervasive health system integrating patient monitoring, status logging, and social sharing enabling self-management of chronic patients in their environment. Junnila, S. et al propose in [111] in-Home health monitoring platform with a “common sensor interface” architecture that support a large number of Zigbee sensors. The platform is tested with two health case studies: one with an elderly woman living in sheltered housing, and the other with a hip surgery patient during his rehabilitation phase.

From medical application side, a middleware for bedsore detection can be found in [112] where authors defined a SOA architecture for bedsores detection and sleep monitoring. The key idea of this work is about collecting information from wireless and wearable sensors. Other monitoring system proposal can be found in [113], it is based on body sensors connected to a smartphone via Bluetooth to get information like heart rate and body temperature.

Uranus [114] is a middleware architecture based on SOA for dependable AAL and vital signs monitoring applications. It provides a rapid-prototype for monitoring of the oxygen in the blood of a chronically ill patient and another prototype for monitoring patients in the context of a smart hospital. MyHealthAssistant [115] presents an event-driven middleware targeted for medical applications on smartphone to enable flexible coupling with changing sets of wireless sensor units. Waluyo et al. [116] propose a middleware for medical Body sensor network that supports multiple sensors and applications, plug and play features, and resource management. In that project, however, parts of the middleware and the applications reside on a single PC.

Message Oriented Middleware: A Message Oriented Middleware has been used for a long time in network communications especially in industrial networks such as integrated manufacturing systems [117] [118]. It offers an event-based architecture and a publish/subscribe communication model. In event-based architecture, components, applications, and all other participants interact through events. Events are propagated from the sending application components to the receiving application components following the publish/subscribe paradigm. The publish/subscribe model is an interaction model that consists both of publishers and subscribers. Data sources (publishers) and destinations (subscribers) are decoupled from each other and data objects (messages) are filtered and delivered to destinations based on predefined topics expressed as subscriptions thanks to a dedicated component called the message broker as depicted in Figure 2.7. The broker can be seen as a mediator responsible for the management of distribution of messages, to serve the right information to the right consumer. The strength of this middleware basically lies in its support for asynchronous communication, allowing a loose coupling between the sender and the receiver.

Since the apparition of the Internet of Things, the publish/subscribe mechanism has been put into light for its effectiveness in offering loose coupling. Compared to the SOA architecture which is also widely proposed for IoT solutions, a MOM follows a message-based distribution model focusing on the information. It differs from the classical client/server paradigm in that neither the source nor the destination of the message have to be known from each other prior to communication.

Few IoT projects have proposed publish/subscribe solutions in the literature. For instance, CenceMe project [119] aims to automatically infer people's activity (e.g.

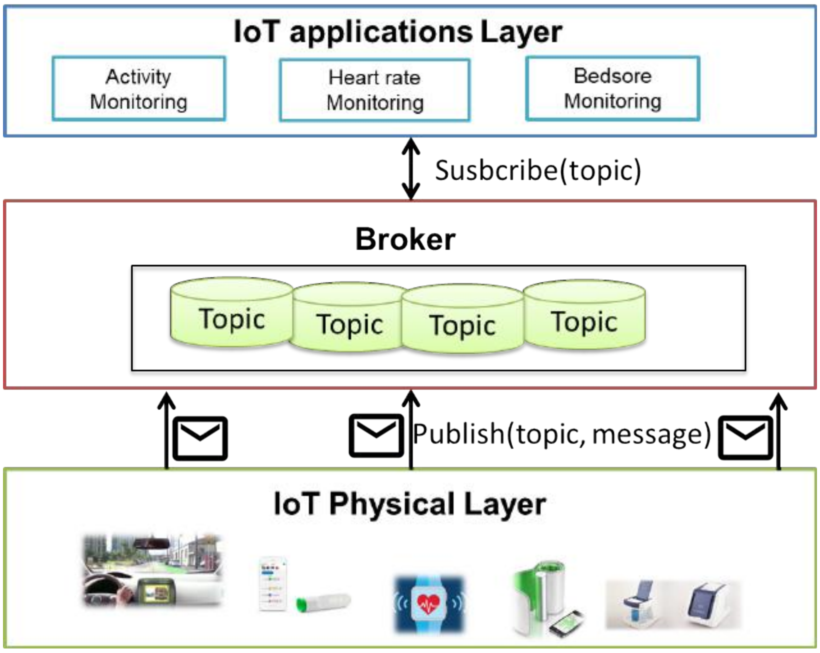


Figure 2.7 General design model for an event-based middleware.

dancing in the party) based on sensor-enabled smart phone in order to share this activity through social media like Facefook. Another example supporting easy access to sensor data on mobile phones is Pogo [120], a publish/subscribe middleware infrastructure for mobile phone sensing. It uses simple topic-based subscriptions to manage access to sensor data and reports significant energy gains due to topic-based filtering of sensed data on mobile devices.

On the other hand a number of Cloud-based services dedicated for storing sensor-based data are nowadays available. Few examples that could be mentioned are Xively [121] ThingSpeak [122] iDigi [123] which support connections using MQTT, they represent a scalable infrastructure that enables users to build IoT products and services, and store, share and discover real-time sensor.

Discussion

Middleware proposals based on Application-specific, Agent-based, VM-based, tuple-spaces and database-oriented, have tried to tackle the requirements that we have stated in section 1.3.1. These approaches have been used in specific IoT applications and their limitations stated in each case make them not so popular in this environment. Moreover, it is clear that the most used approach in the IoT world is SOA-based architecture. However, we recognized that even if SOA presents a good solution for

IoT, it can not meet all IoT requirements especially related to heterogeneity, scalability, real-time processing and flexibility. From the other side, we recognized that although few recent studies have considered the MoM approach, it has been recognized as flexible and suitable communication model in IoT. Based on the before mentioned middleware solutions, a comparison between SOA and MoM shows that:

- In regards to SOA architecture, MoM follows a message-based model focusing on the information itself and supports sending and receiving of messages between distributed systems. While in SOA approach, data providers and data consumers agree on a service contract before data flow sharing as it is clear in Figure 2.8, this approach makes the communication not completely decoupled between system components.
- Despite of its historical role in sensors management and its potential benefits in IoT middleware solutions, SOA-based solutions focusing on services provided by the system do not scale well in ultra large and dynamic IoT environments. While on the other side, publish-subscribe approaches are gaining more momentum due to their ability to promote scalable, flexible and fully decoupled communication in intelligent environments. As a result, we believe that a MoM with publish/-subscribe paradigm is the most suitable middleware solution for IoT healthcare applications.

To sum up, middleware solutions have been found to address technical interoperability issues and communication requirements in an IoT environment. However, the heterogeneity of connected devices and the variety of data sent over the network introduce the requirement of providing a unified and coherent manner of representing data among system components. As well as middleware approaches aim to hide the technological details, virtualizing hardware and technological infrastructure by providing a semantic description would enhance the interoperability and provide a full abstraction of the technical level. In the following section, we present how semantic technologies have been integrated in IoT.

2.4 Semantic Description in IoT

Web of Things [124] is an improvement of IoT allowing an easier way for IoT applications to build upon smart things. The Web of Things (WoT) concept relies on the

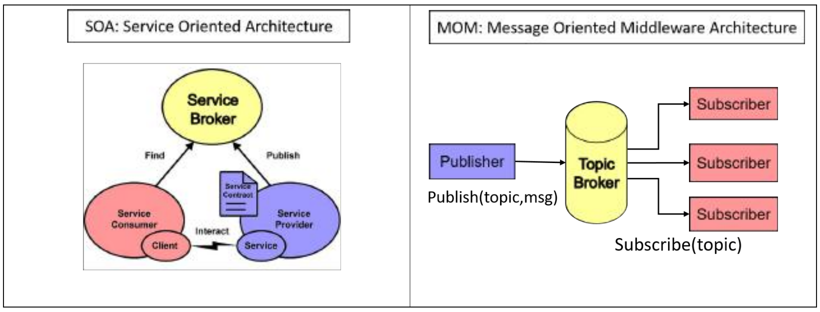


Figure 2.8 Service Oriented Architecture vs Message Oriented Middleware Architecture

connectivity service of IoT and easy access to sensors in order to create applications exploiting the IoT data [125]. Moreover, using semantic description allows data contextualization for optimized data stream discovery, indexing and querying. Although, new research projects are shifting to semantic web of things [126] where data can be integrated with data and services available in other information systems. This flexibility facilitates the production of novel applications and services that are based on the state of the real world. It also supports autonomous semantic reasoning and decision making mechanisms to provide higher-level actionable knowledge from low-level sensor data [127]. Hence, performing semantic reasoning is linked to the ability to define a description model of sensors observations.

In the research area, some projects have addressed IoT heterogeneity challenges from a semantic perspective and they found that a global and scalable understanding of IoT services syntax as well as semantic description is still required for a robust and interoperable system. Semantic representations have been widely discussed by IoT researchers from different perspectives that we develop in this section: sensors modeling and description, exchanged data description and domain description.

2.4.1 Sensors modeling and description

Sensors description has been proposed from one hand to promote reusability and integration of sensors and from the other hand to help solving the difficulties of installing, querying [18] and maintaining complex, heterogeneous sensor networks. In this field, XML schemas have been used for sensors description such in sensor model language (SensorML) [128] that provides metadata model to describe sensor capabilities and measurement process. They are also used by SOA architecture to describe sensors characteristics as a service contract to data consumers. Also, ontologies [129] for sensors description have been widely used as a reliable technique due to its considerable efficiency in ensuring both interoperability and clarification of knowledge structure.

The main advantage of ontologies is their ability to describe three perspectives of a sensor [130]. First, it describes sensors metadata, sensor type, components, configuration, process and properties. Second, it allows to describe what a sensor can measure after stimulus detection or the observation of the sensor in term of frequency, accuracy and measurement capabilities. And third, it provides the concepts to describe data stream and the observation result. Authors in [131] present a general state of art of sensor network ontologies and extract the main use of ontologies. It can be summarized as follows:

- Develop a vocabulary;
- Define an interchange format to facilitate data transfer;
- Describe sensors or manipulate sensor data;
- Integrate data to access uniformly to heterogeneous data sources;
- Generate new knowledge;
- Detects inconsistency;

2.4.1.1 Overview of the Existent Ontologies for sensors description

Key standardization efforts that have sought to establish sensor data models for sensors to be accessible and controllable via the Web include the OGC Sensor Web Enablement (SWE). The SWE efforts established by the Open Geospatial Consortium include following important specifications: Observation & Measurement (O&M), Sensor Model Language (SensorML) and Sensor Observation Service (SOS) [132]. The O&M and SensorML contain standard model and XML schema for observations/measurements and sensors/processes respectively. The SOS is a standard service model, which provides mechanism for querying observation and sensor metadata. Based on these standards, several ontology approaches have been developed to describe sensors.

Compton et al. [133] and Bendadouche et al. [134] have surveyed the several attempts of ontologies aiming at establishing a sensor ontology covering the description of all sensor topics. From 2004 to 2008, ontology projects for sensors description have not been complete and they have been discontinuous however, these attempts have helped to build other ontologies. These projects refer to the ontology in [135] where authors describe an ontology for adaptive sensor networks where nodes react to available power and environmental factors, calibrating for accuracy and determining suitable operating states. The OntoSensor [136], a comprehensive sensor ontology based on SensorML, was intended as a general knowledge base of sensors for query and inference. Matheus et al. [137] include sensor types in an ontology developed for recording provenance, or pedigree, information in naval operations. It allows to find a sensor after a request specifying the provided services. Kim et al. [138] extend OntoSensor for Web services, though the ontology or full details are not available as well as the ontology proposed by the authors in [139].

In 2008, four potential ontologies have been proposed CESN, SWAMO, A3ME and ISTAR. The Coastal Environmental Sensing Networks (CESN) [140] project for sensor networks for coastal observing has built an ontology to describe relationships between sensors and their measurements. It also provides logic programming rules reasoner to validate sensor observation and to test anomalous sensor observation by decision maker. SWAMO [141] describes physical devices and process models and tasks in a distributed and intelligent software agents environment. And A3ME ontology [142] was developed to classify devices and their capabilities in a heterogeneous network. The ISTAR [143] ontology was developed as part of a system to automatically select sensors for tasks based on their fitness for the task description.

In 2009, OOSTETHYS [144], MMI [145] and CSIRO [146] ontologies have been proposed. The OOSTETHYS ontology has been designed to describe sensors and get observation and capabilities for oceanographic observing. Likewise, The Marine Metadata Interoperability (MMI) Device Ontologies Working Group has developed the MMI ontology of oceanographic devices, sensors and samplers. The CSIRO ontology is a generic ontology for describing sensors and deployments. It is intended to be used in data integration, search and classification features. It can expresses complex compositions and finds details of the function and results of sensors and processes. The ontology can also encode much of the information in SensorML documents.

Ontologies developed until 2009 are either specific to a domain (oceanographic, ecology, etc.) or discontinued, therefore these efforts did not lead to a mature and general solution applied for ongoing projects. None of the ontologies is able to express all the properties required for a full description of sensors and a standard is still not available until 2011.

In 2011, the W3C organism proposed SSN, the Semantic Sensor Network Ontology [20] represented in Figure. 2.9. It has been initiated by the CSIRO, Wright State University, and the OGC as a forum for the development of an OWL ontology for sensors and to further investigate annotation of existing concepts, and links to existing standards. SSN has been designed as a generic ontology which has become one of the most popular and efficient ontology to describe sensors and observations. It has been conceived as a domain-independent ontology where extensions can be made to add domain-specific knowledges.

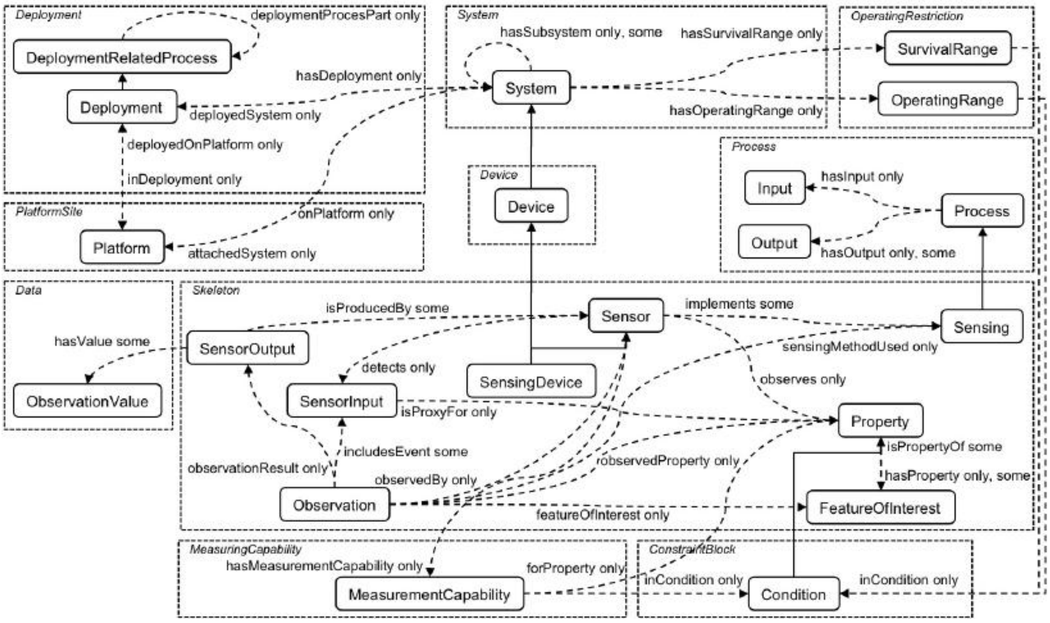


Figure 2.9 Semantic sensor Network (SSN) Ontology design

2.4.1.2 Semantic Sensor Network Ontology Extensions

Extensions of SSN have been developed in the last years by adding concepts related to time, space, communication and concepts related to domain characteristics. We review in this section a broad range of these solutions considering that the presented ontologies are just a small example of existing ontologies that are built on top of SSN. It helps showing the impact of SSN on the semantic description of sensors over the years.

BFO [147] is a spatio-temporal extension that make distinction between describing identities that happen at a finite time and events like storm and routing. It is a hierarchical system approach where sensors collect data from real world and send it to clusterhead-node.

Bandadouche et al. in [134] presents the limitation of SSN in describing the communication process of wireless sensor network. They propose a new ontology design pattern Stimulus-WSNnode-Communication, an extension of SSN, that addresses the communication limitation by integrating new concepts that describe the communication process of wireless sensor network. Another extension for SSN based on fuzzy logic is proposed in [148] to support fault tolerance and for large scale Wireless Sensor Network. It is a service oriented approach to build diagnosis and test services for wireless sensors.

Authors in [149] present another SSN extension ontology for WSN and it is presented as alignment of many ontologies SSN, SWRLTO, TAO and DOLCE. The aim of this solution is to improve time description's limitation in SSN by representing temporal abstraction to analyze data in real time. SSN is used for sensor's measurements; SWRLTO is used for temporal modelling and reasoning; and TAO designed by the authors of this paper to capture the semantic TA (Temporal Abstractions). This framework uses temporal reasoning to search and classify temporal patterns that help to infer the processed data. For the alignment of the three ontologies the authors use DOLCE, a known upper ontology.

Recently, in [150], authors have analyzed general IoT ontologies: SSN, Smart Appliance REFerence (SAREF) [2], iot-ontology [3] , IoT-lite [4], Spitfire [5] , IoT-S [6], SA [7] and the oneM2M ontology [8]. They did not consider specific domains impacted by IoT (domotics, agriculture, smart cities...) but they studied ontologies that they found on the web. Based on their comparative study, none of the presented ontologies can describe actuators. Actuators are devices that transform an input signal into a physical output, making them the exact opposite of sensors. They have proposed IoT-O ontology in order to describe actuators, services and energy consumption.

To sum up, all aforementioned ontologies aim at describing physical sensor data and meta data. However, no one of the ontologies describes virtual sensors that can

[2] http://sites.google.com/site/smartappliancesproject/ontologies

[3] http://ai-group.ds.unipi.gr/kotis/ontologies/IoT-ontology

[4] http://iot.ee.surrey.ac.uk/fiware/ontologies/iot-lite

[5] http://sensormeasurement.appspot.com/ont/sensor/spitfire.owl

[6] http://personal.ee.surrey.ac.uk/Personal/P.Barnaghi/ontology/OWL-IoT-S.owl

[7] http://sensormeasurement.appspot.com/ont/sensor/hachem onto.owl

[8] http://www.onem2m.org/ontology/Base Ontology/

be presented in an IoT system. We recognize virtual sensors as any application that generates information that can not be gathered from physical world. For example, an application that uses motion sensor in a living room data and outputs the presence of the person can be seen as virtual sensor in IoT world.

SSN is commonly used by many projects and is still the most appropriate ontology that can describe an IoT sensor system. Describing physical sensors with SSN promotes the reusability of sensors and makes IoT systems sustainable. However, SSN should be extended by concepts to describe virtual sensors.

2.4.2 Web Ontology Language for Data Description

Sensor data in smart environments are published from different data providers and received by many data consumers as we have shown in section 2.2.2. The huge number of heterogeneous sensors with specific characteristics make difficult for applications to query sensors and share information. To address this issue, research initiatives and standardization activities have mainly focused on modeling sensors observations and on sending the semantic observations over the network.

In [151], different data models have been presented as coherent and reliable way to share information across an IoT system: XML, Web Service Definition Language (WSDL), JavaScript Object Notation (JSON), Resource Description Framework (RDF) and Web Ontology Language (OWL).

WSDL is a description interface that use XML document to store this description. This approach is mainly used in service oriented architectures for IoT such as [152]. JSON is a simple data description that also has been used by several IoT projects such as [153]. Whereas, RDF and its new version OWL format has become the most used technique for sensor data description due to their ability to facilitate intelligent functions in IoT including reasoning over sensor data and semantic interoperability among devices [154] [155]. Moreover, RDF/OWL languages are generated by means of ontologies technology.

The Web Ontology Language (OWL) built upon a W3C XML standard is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. It can easily be integrated in many programming languages like Java through APIs (Jena or OWL-API). Beside the important role of ontologies in representing sensors networks, physical aspect and infrastructure, a main advantage of using them is the possibility to generate OWL representations.

It also provides the opportunity to exchange formal messages at the technical layer in OWL format that can guarantee the reliability of information and improve the interoperability of the system. Finally, it fills the need to fuse heterogeneous sensor data, and potentially inaccurate data.

2.4.3 Ontologies for Domain Description

Domain-specific IoT applications are becoming increasingly popular and domain knowledge description is gaining even more popularity. Domain knowledge is already defined in more than 200 ontologies and sensor-based projects [156]. These ontologies provide expert knowledge description as well as conceptualization of the different domains like transportation or diseases. Specifically, domain applications use ontologies and semantic description as a way to define types, properties, and interrelationships of the entities that exist for a specific domain.

As an example, authors in [157] have proposed an ontology to define logistic terminologies. It is intended to assist a logistics expert to identify and precisely specify the logistic problem for solving passenger train optimization problem. In the same context, in [158], an ontology is proposed for monitoring and improving public transportation. Another example of ontologies domain is related to energy concepts as presented in [159] where authors created ontologies to describe energy sources: wind power, biomass, and fossil fuels. Further domains have relied on ontologies to describe concepts and relations that constitute their particular fields. In agriculture we distinguish AgroPortal ontology for agriculture and nutrition data [160] as well as the proposal in [161] and AgOnt [162].

In healthcare, many ontologies describe diseases. For instance, the Disease Ontology [163] represents human diseases for linking biomedical knowledge through disease data. Also, the well known SNOMED-CT [164] represents an advanced terminology and coding system for eHealth. In the same context, author in [165] proposed an ontology to describe the patient's vital signs. The aim of this ontology is to create a profile for each patient containing the several medical control information. Moreover, author in [166] present an example of food ontology for diabetes control. The HL7 Security and Privacy Ontology is another example that serves to name, define, formally describe, and interrelate key concepts within the scope of Healthcare Information Technology. Kim et al. [167] proposed an Ontology driven Interactive Healthcare with Wearable Sensors (OdIH_WS) to achieve customized healthcare service. It aims to acquire context information at real-time using ontological methods by integrating external data such as meteorological web site in order to prevent disease. ContoExam [168] is an ontology developed to address the interoperability problem of sensor networks in the context of e-health domain applications. It contains specific expressions and specifications for medical use as examination vocabulary and expressions.

Discussion

In this section, we have discussed the several perspectives of using semantic description in IoT. We have found that ontologies are the most common semantic technology that describe the three main concepts: sensors, data and domain.

A well designed ontology enhances IoT environments and architectures by ensuring some features like abstraction, scalability, integration, smart reasoning and interoperability as summarized in Figure 2.10. Moreover, an IoT ontology model relies on several design principles: it should be modular to facilitate its evolution, extension and integration with external ontologies; and it should be lightweight to be widely adopted and reusable. Furthermore, an ontology should be a complete reflection of a full description of an IoT environment which reduces the need to import other ontologies. The compatibility principle is related to the needs to be consistent with existing ontologies.

We have shown that SSN is a suitable ontology for sensors description in IoT and is able to generate data in RDF/OWL format. Presenting data in a formal way using RDF/OWL enhances the interoperability and homogeneity of information between IoT systems. In the context of healthcare application, several ontologies have been proposed to describe diseases and domain concepts.

However, sensors description and domain description have always been developed apart from each other. There is no ontology proposal that is able to combine between sensors and domain concepts in order to provide a full description in IoT.

2.5 Ongoing Challenges and Open Issues

In this chapter, we have presented a broad range of the proposed solutions in the healthcare domain. We found that most solutions focus on information gathered from sensors and they do not put attention on the way of extracting these information. Their main objective is to provide clinical validation and individual prototypes of diseases monitoring applications. Moreover, these systems focus on a single disease or pathology whereas elderly people are commonly subject to several chronic pathologies at the

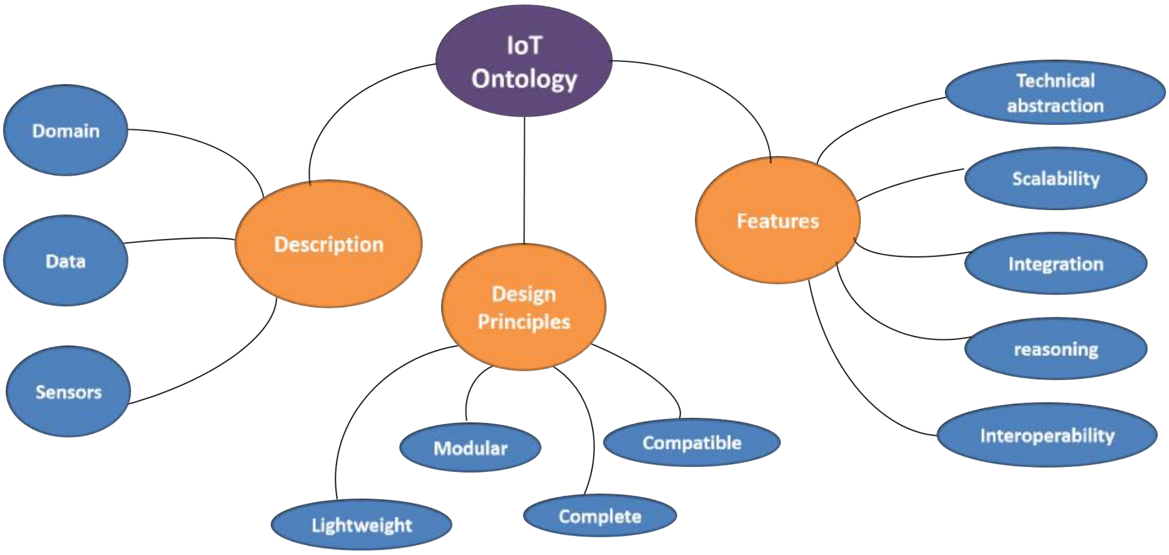


Figure 2.10 Main characteristics of an IoT Ontology

same time. Therefore, it becomes cumbersome to use several unrelated systems to cope with every diseases/affections. In such case, the number of redundant sensors may be annoying and disturbing for patients. Although, we recognized a lack of software engineering methods that can manage heterogeneous devices and several applications.

Middleware is a suitable solution to manage communication between physical sensors and IoT applications. The proposals are diverse and involve various middleware design and support different requirements to operate in IoT domain. We chose the MoM model to cover our IoT communication approach. However, it is not sufficient to provide a full homogeneous IoT system. Some requirements such as scalability, reliability, interoperability and integration remain open.

From the other side, it is assumed that detection solutions use a set of ad-hoc sensors, and their architectures differ deeply from one another. Thus, the number and variety of sensors under use can be very important. If we take the example of a monitoring application where we need a temperature sensor for a risk assessment calculation. If the temperature sensor should be replaced by another one for a specific reason, observation sent from the 2 sensors can be either in Celsius or in Fahrenheit, in a different accuracy and different frequency which will impact the results. This heterogeneity complicates sensor data sharing and replacement of sensors. Therefore physical sensors should be at a low level, and a semantic representation of sensor data would be a mandatory replacement at the first stage.

We have shown that sensors description as well as data description promotes the interoperability between heterogeneous components, and hence are capable of spreading homogeneity of sensors information in an IoT system. SSN is a suitable ontology to describe sensor data and metadata however, it lacks from describing virtual sensors that generate analyzed data to the network. Virtual sensors can add domain knowledge to the same ontology which does not exist in other ontologies.

2.6 Conclusion of the Chapter

The state of the art presented in this chapter has covered the three main keys of our research, middleware architectures, semantic techniques and healthcare applications. Our analysis shows that middleware can be appropriate to manage IoT systems however, a global and scalable understanding of IoT data syntax and semantics is still required.

As a conclusion, proposed solutions in the literature can be relevant for specific applications and do not cover all technical and semantic challenges in IoT middleware. We believe that adding semantic technologies over a middleware model could offer a solution with high interoperability and flexibility for IoT applications.

Therefore, a new lightweight semantic middleware architecture that can provide an effortless programming environment is mandatory to exchange health related semantic data. In the previous sections, we have argued the choice of MOM architecture with a publish/subscribe paradigm that complies with our requirements and focuses on information rather than services. From the other side, SSN ontology for sensors and observation description present a prevalent knowledge description technique. However, it lacks concepts to describe virtual sensors and domain knowledge. Therefore, we propose to design a new complete ontology CoSSN. We chose to rely on SSN ontology with the addition of an extension allowing full semantic description in healthcare IoT system.

The combination of MOM and CoSSN will be the pillars of our strategy to propose a new approach of IoT semantic middleware, SeMoM, for flexible, interoperable and scalable communications between IoT components.

Chapter 3

CoSSN: The COgnitive Semantic Sensor Network Ontology

> "All our work, our whole life is a matter of semantics, because words are the tools with which we work, the material out of which laws are made, out of which the Constitution was written. Everything depends on our understanding of them."
>
> — *Felix Frankfurter*

Contents

3.1 Introduction

In this dissertation, we propose a semantic IoT architecture to develop domain applications able to exchange semantic messages. From a semantic perspective, our objective is to provide a full description of sensors and domain concepts, merging the cyber-physical world, the real world and the expert knowledge. This combination is prone to provide an interoperable and flexible data sharing and knowledge processing between system components as discussed in section 2.5. Our methodology is based on ontologies as semantic modeling method for IoT description regarding its important role in improving the semantic interoperability as described in section 2.4. Moreover, ontologies enable semantic reasoning through IoT system which makes the system smarter and efficient by inferring new knowledge for decision making. The aim of this chapter is to present our approach for the construction of a sensor ontology model adapted to IoT world and AAL environment. Our approach relies on the well known SSN Ontology that has been extended with more comprehensive concepts to tackle IoT healthcare applications requirements.

We present first in section 3.2 our vision in knowledge and data description in IoT and we put a light on the SSN Ontology which is the nexus of our proposal. We define and describe in section 3.3 a sensor hierarchy used to identify the different types of sensors presented in the network. A data model and data characterization is also presented and describes sensors observations characteristics. We then present in 3.4 COSSN, our approach in designing an ontology-based model for sensors and knowledge description in smart environments. A focus on querying semantic data is then presented in section 3.5 showing the process to retrieve the relevant information from sensors. We present in section 3.6 the bedsore risk detection as a medical application case study. This example will be used as to illustrate our semantic approach. Finally, we discuss our proposal and present the advantages and limitations considering the other proposals in the literature before concluding the chapter in section 3.7.

3.2 Knowledge and Data Description in IoT

Virtualization promotes the detachment from the technical deployment of sensors such as location, storage structure, access format, and streaming technology [169]. Generating semantic description of sensors is a form of virtualization that brings abstraction of sensors characterization and technology. We present the SSN ontology which is a semantic model for sensor abstraction and characterization. It complies to the basic need of sensors network modeling and employs an ontology design pattern (ODP) which serves as root for our semantic methodology.

3.2.1 SSN: The Semantic Sensor Network Ontology

SSN ontology (Semantic sensor Network ontology) [20] is a generic and domain-independent ontology introduced by the W3C in 2011 to model sensor devices, systems and processes. It is aligned with the DOLCE Ultra Lite (DUL) foundational ontology [1]. DUL is an upper ontology with the aim to provide a set of upper level concepts that can be the basis for easier interoperability among many middle and lower level ontologies.

SSN ontology is compatible and compliant with OASIS (OASIS Group Consortium) standards at the sensors and observations levels whereas, concepts and properties are related to SensorML and O&M standards. SSN is organised conceptually in 10 modules: Deployment, System, Operating restriction, Platform site, Device, Process, Data, SSO pattern, MeasuringCapability and ConstraintBlock. The full ontology consists of 41 concepts inheriting from 11 DUL concepts and 39 objects properties inheriting from 14 DUL object properties. It allows the accurate description of sensors and allows the introduction of new domains with the addition of dedicated vocabularies. In an IoT system, an application is interested mainly on sensor observation and data value. These information are described in the SSN ontology with its main concepts Observation, SensorOutput and Property (see figure. 3.1).

[1] http://www.ontologydesignpatterns.org/ont/dul/DUL.owl. Last access on August, 2017.

SSN can be seen from four main perspectives: a sensor perspective, an observation perspective, a system perspective, and a feature and property perspective.

- Sensor perspective: focusing on how sensing is performed and on what is sensed. In other words, it describes which sensation detected by which device and how it is sensed;
- Observation perspective: focusing on observation data and their metadata;
- System perspective: focusing on sensor network and its deployments;
- Feature and property perspective: focusing on the relation between concepts, it allows describing which sensation is sensed by which property and for which observation.

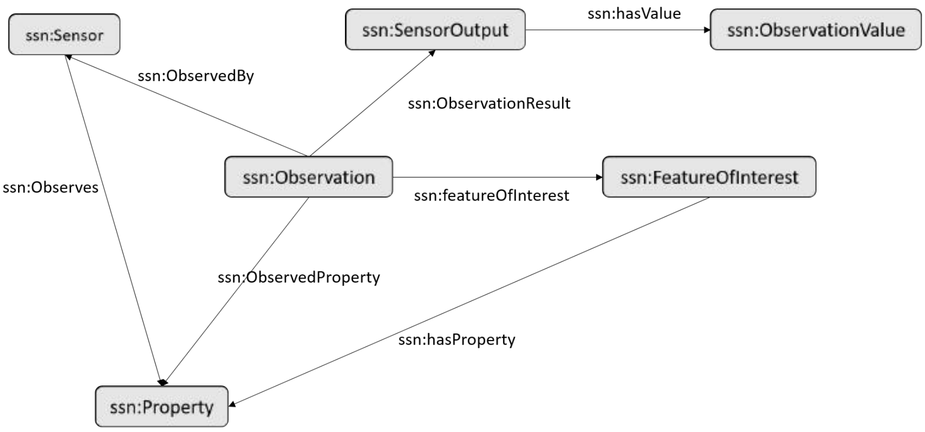


Figure 3.1 Important concepts of SSN

Stimulus-Sensor-Observation Design Pattern

SSN is built around a sensor Ontology Design Pattern (ODP) which is called SSO. It describes relations between sensors, stimulus and observations “Stimulus-Sensor-Observation” [170]. The SSO in Figure 3.2 has been developed following the principle of a minimal, common ground for heavy-weight ontologies for the Semantic Sensor Web to make it reusable for a variety of application areas, as well as to explicitly address the need for light-weight semantics for Linked Data. In the following, we describe the concepts and relations that jointly form the SSO pattern.

Stimuli: are detectable changes (dul:Event) in the environment and act as a trigger for sensor and a way to measure a property. Nevertheless, a stimulus (ssn:Stimulus) may only be usable as proxy (ssn:isProxyfor) for an observable property (ssn:Property). For example, changes in electrical resistance are the "stimuli" that trigger a thermistor that observes "temperature" property. Another example, the current of air are stimuli generated by spinning wind cups for wind speed "property" observed by wind sensor. Properties themselves are observable characteristics of (ssn:isPropertyOf) real-world entities (ssn:FeatureOfInterest).

Sensors: sensors (ssn:Sensor) are physical objects that perform observations, i.e., they transform an incoming stimulus (ssn:detects) into another, often digital, representation (ssn:SensorOutput). Sensors may be hardware devices, sensing systems, scientific computational models, human run laboratory setups – anything that senses. Furthermore, there is distinction between the sensor and a procedure, i.e., a description, which defines how a sensor should be realized and deployed to measure a certain observable property. This may be, for example, a description of the scientific method implemented by the sensor. The sensing method, though, is distinct from a process or workflow description of how the sensor operates (not described by the SSN ontology, workflow and process descriptions being more widely applicable and expected to be imported from a suitable ontology).

Observations: observations are either modeled as database records that store the sensor output and additional metadata or as events in the physical world. While in SSO, they are defined as social constructs (ssn:Observation). For a sensing event, an observation can link the act of sensing (dul:includesEvent), the event that is the stimulus (dul:includesEvent), the sensor (ssn:observedBy), a method (ssn:sensingMethodUsed), a result (ssn:observation-Result), an observed feature (ssn:featureOfInterest), and property (ssn:observedProperty), placing all in an interpretative context including parameters such as time and location.

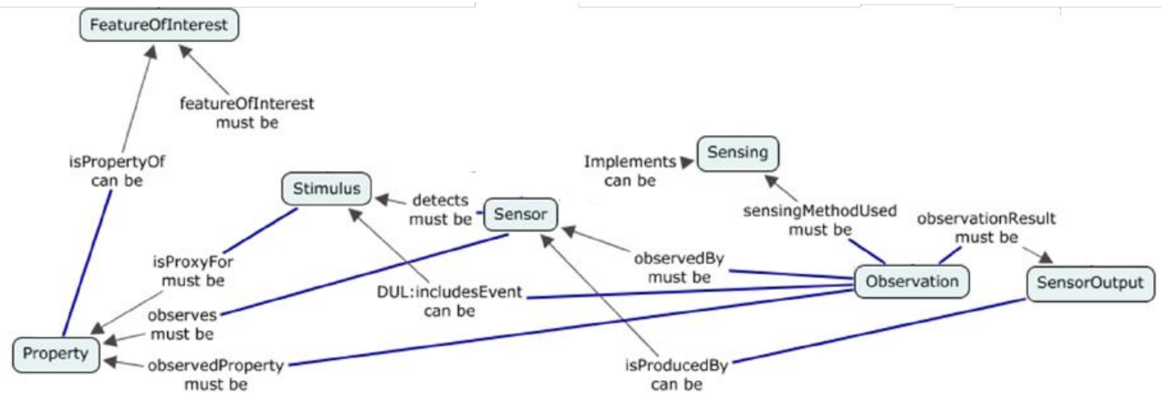


Figure 3.2 A conceptual map showing the core concepts and relations forming the Stimulus-Sensor-Observation ontology design pattern.

3.2.2 Ontology-based Methodology for knowledge Description

As described in SSN, ontologies are mainly developed following an Ontology Design Pattern (ODP). The term "design pattern" has been introduced by Christopher Alexander in [171] for shared guidelines that help solving design problems. A good design can be achieved by means of a set of rules that are packaged in the form of patterns. Hence, design patterns are assumed as archetypal solutions to design problems in a certain context. More recently, design patterns also appeared in requirements analysis, conceptual modeling, and ontology engineering [172], called Ontology Design Pattern (ODP). Developing ODP for ontology design is known as quite complex, many proposals have introduced methodologies to design high quality and scalable ontology such as [173] [174] [21]. Our new ontology approach extends SSN ontology and follows also the principles of SSO ontology design pattern in designing new concepts related to the domain requirements.

To use our approach, an IoT architect follows a semantic methodology presented in Figure. 3.3. It consists first on defining the deployed sensors with their characteristics and the threshold values needed to perform a specific calculation. Second, sensors are described using our ontology approach. After that, semantic sensors share and communicate their semantic information to the appropriate IoT applications in the third step. Finally, applications apply reasoning techniques and infer new knowledge. In this step, it is possible to discover new needs of sensor data then, the architect repeat the same steps starting from defining these sensors. This approach ensures five main tasks emphasized and detailed in the following sections. These tasks refer to model sensor data, model experts knowledge, retrieving sensors output, inferring knowledge from sensors output and finally searching for existing sensors in the deployed IoT network.

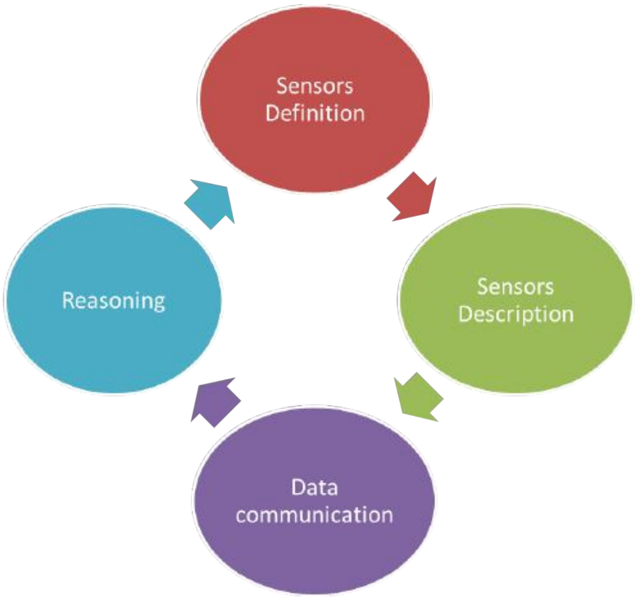


Figure 3.3 Methodology cycle of modeling bedsore detection application with CoSSN

3.3 Sensors modeling and Data characterization

IoT based applications are becoming increasingly demanding. In fact, an IoT system should offer to applications the possibility to access to any sensor, any information and any context at anytime, anywhere and related to any person [175]. To answer to application requirements, new applications and intelligent sensors will integrate an IoT system aiming at contextualizing or generating high level of information. Thus, in an IoT system physical sensors will not remain the only data source. We will have connected applications and virtual sensors that are able to send new observations with enhanced knowledge.

Applications requirements are then making the definition of the shared data between sensors, hence the description of the relative sensor data more complex. In order to find a formal description of these shared information, it is cumbersome to define the different types of sensors that can be presented in an IoT environment today. Sensors hierarchy and relation between different components are also necessary to define formal description of sensors. In the following we present the smart sensors hierarchy, our approach in designing sensor model and the correspondent data model for IoT monitoring systems.

3.3.1 Definition of a Sensor Hierarchy

Sensors and connected objects need to be described and formalized in the way that different applications can use the same sensor and same information. Ontologies offer a formal model to represent different levels of abstraction with rich semantic description that can be used for reasoning. We propose a definition of sensors hierarchy that rely on the level of the semantic enrichment of sensors and also on the level of knowledge that a sensor is capable to spread in the network.

As sensors are considered smart objects, in our hierarchy the smartness of sensors is also related to the level of richness of semantic data and the level of knowledge provided by each sensor. Based on these characterization factors, we define a sensor hierarchy of three main connected sensors as depicted in Figure. 3.4: physical sensor, semantic sensor and virtual semantic sensor. We define also the cognitive sensor as a type of virtual semantic sensor with more specifications. Physical and semantic sensor are domain-independent sensors however, the virtual semantic sensor and cognitive sensors will be specialized sensors description for a specific domain.

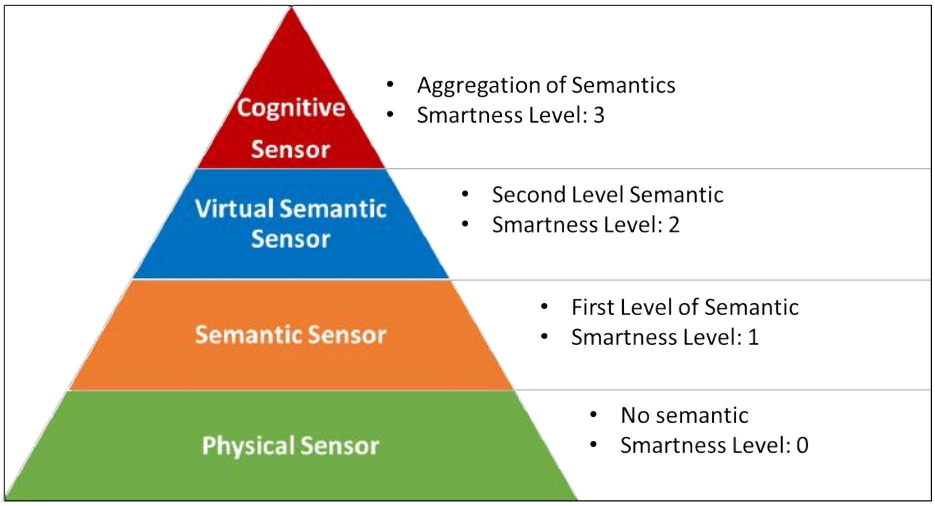


Figure 3.4 Smart Sensor Hierarchy and the relative Smartness levels

3.3.1.1 Physical sensor

Physical sensor is the pillar of the whole hierarchy. It is the main data source of any IoT system. There are no semantic rules applied at this level and the smartness level is 0. In fact, in our approach physical sensors will not share their raw data in the network since the main concept of our approach is to exchange formal and semantic data over the network.

Physical sensor [176] is a device that detects a physical property and records, indicates, or otherwise responds to it. Physical property of a sensor is the fundamental for measuring physical quantities and converting them into signals which can be read by an observer, taking into account that the output signal is determined by the input signal.

The specific input could be light, heat, motion, moisture, pressure, or any one of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing. For example, a temperature sensor detects temperature property from real world and sends the signal necessary to read a numeric value like "25".

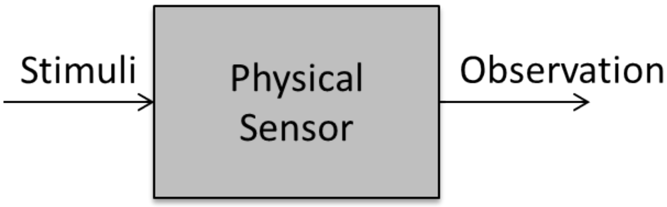


Figure 3.5 Phyical Sensor

Physical sensors are referred to smart connected devices in an IoT system. They represent any entity that detects physical stimulus and sends sensor observations. It can have many properties like accuracy, observation time, frequency, feature of interest, classification, unit of measurement etc. IoT applications connect to the network with the aim to receive the necessary physical sensors observations for its reasoning and calculation. For that, a preliminary study of potential physical sensors needed for its calculation is required to define the corresponding threshold values necessary for calculation.

3.3.1.2 Semantic sensor

One of the most critical requirements of the IoT is that of making heterogeneous objects plug-and-playable and interoperable. As soon as an object joins a network, it needs to be immediately provided with mechanisms that enable interaction with the external world. One of the most effective and efficient solution to represent IoT objects is by using semantic technologies [177]. Consequently, adding more knowledge and information to the physical sensor is then an effective and consistent way to improve the precision of sensor data.

Ontologies refer to semantic mechanisms, coherent and unambiguous knowledge representation. It also refers to an abstract conceptual vision of the world and of sensors. Therefore in our approach, the semantic description of sensors by an ontology allows to identify the semantic sensor hence, to describe physical sensor data and metadata.

Based on that, we define a **semantic sensor** as a software component that converts raw data into semantic data. Using SSN Ontology, this component generates a semantic description of physical sensor metadata as well as sensors observations. It is responsible for wrapping sensors observations in an OWL messages before sending it as semantic observations. A semantic sensor is a domain-independent description of a physical sensor, information presented at this level are relevant to any IoT domain application. The **semantic sensor** presents the second level of the sensors hierarchy with the level 1 of smartness. The semantic sensor can be viewed as a main data source in our approach.

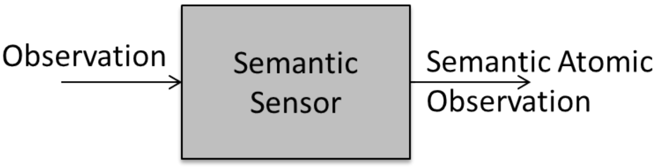


Figure 3.6 Semantic Sensor

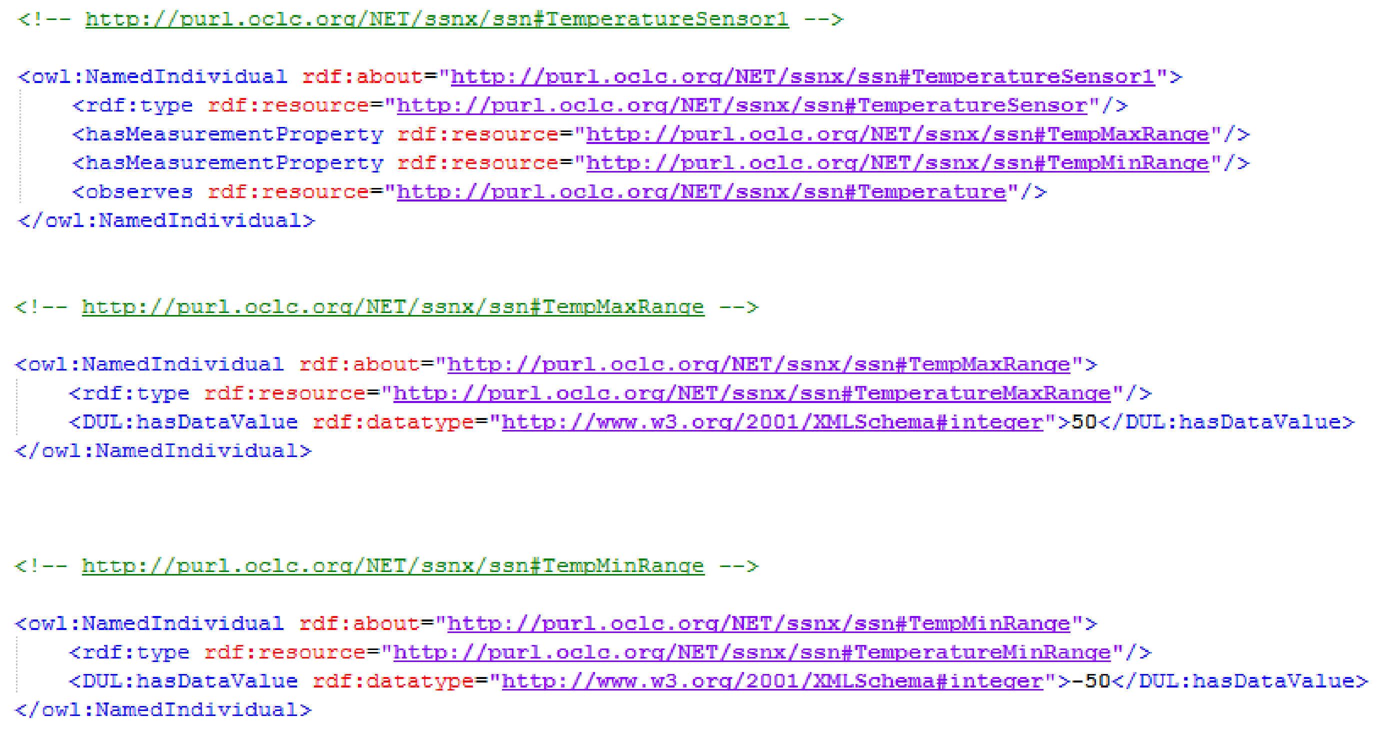
For example, if we deploy physical sensors such as temperature, humidity, pressure and pedometer in a smart environment. In our approach, each one of these sensors is represented as a semantic sensor. Semantic sensor detects sensor data as stimuli and generates semantic observations. More concretely, a semantic temperature sensor receives raw data like temperature = 20. It wraps it in an OWL observation containing the following information: data value is 20 with unit of measurement Celsius for temperature **property**, the observation time is 7:00. Same for the other physical sensors, observations are wrapped and sent as semantic observations. It should be noticed that, sensors property like "temperature" is very important and necessary in semantic sensor definition. In fact, a property can be seen as a label of observations that can be queried based on the property.

Example of Semantic Temperature Sensor with SSN

Figure 3.7 presents a part of a semantic temperature sensor designed with SSN which can be modeled via Protégé software. Based on this description, it can be seen that *TemperatureSensor1* is defined as an instance of class "TemperatureSensor" which is a type defined as a subclass of "Sensor Device" in SSN. The minimum and maximum operating values in the measurement capability of the sensor are also described. Hence, the described sensor operates in a range of values between -50 and 50 Celsius. The property *Temperature* is modeled via the *observes* object property.

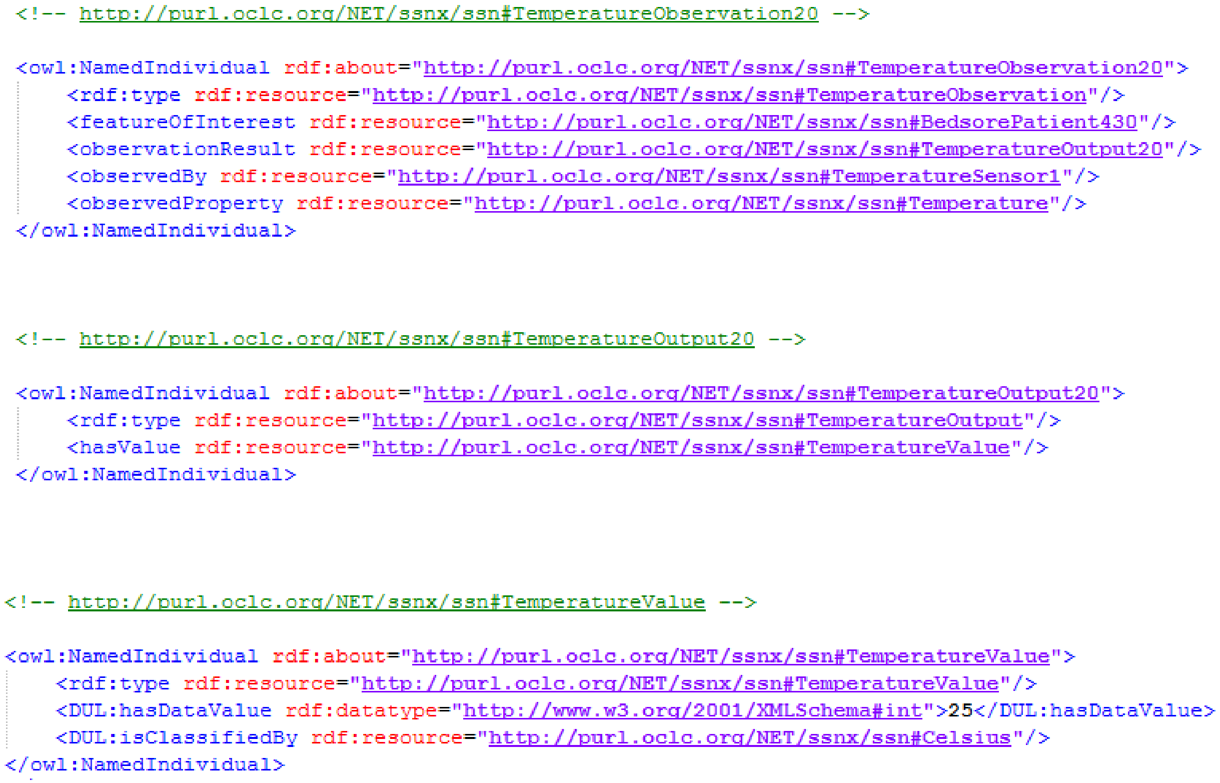
In an IoT system, observations and sensor output present the main target of connected applications which request these data to perform real-time analysis and make diagnosis and decisions based on it. For that, modeling observations and sensors output is crucial since it will be the unique shared information between system components. An example of a temperature observation is presented in Figure. 3.8. Each individual observation defines a feature of interest (i.e., BedsorePatient430), an observation result (i.e, TemperatureOutput20), the sensor that has generated the observation (i.e, TemperatureSensor1) and the property *Temperature*. The sensorOutput presents the value "25" Celsius.

To sum up, semantic sensors offer a semantic representation of sensors raw data that reflect the real physical world. The broad information generated by this semantic sensor could be useful for general statistics. However, for healthcare applications more precision and details are required to monitor several persons with several diseases. Our semantic approach handles this need through a new component and data provider in an IoT system, the virtual semantic sensor. It will be able to describe processed data and knowledge information that can not be detected in real physical world.



```
<!-- http://purl.oclc.org/NET/ssnx/ssn#TemperatureSensor1 -->

<owl:NamedIndividual rdf:about="http://purl.oclc.org/NET/ssnx/ssn#TemperatureSensor1">
    <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureSensor"/>
    <hasMeasurementProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TempMaxRange"/>
    <hasMeasurementProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TempMinRange"/>
    <observes rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#Temperature"/>
</owl:NamedIndividual>


<!-- http://purl.oclc.org/NET/ssnx/ssn#TempMaxRange -->

<owl:NamedIndividual rdf:about="http://purl.oclc.org/NET/ssnx/ssn#TempMaxRange">
    <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureMaxRange"/>
    <DUL:hasDataValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">50</DUL:hasDataValue>
</owl:NamedIndividual>



<!-- http://purl.oclc.org/NET/ssnx/ssn#TempMinRange -->

<owl:NamedIndividual rdf:about="http://purl.oclc.org/NET/ssnx/ssn#TempMinRange">
    <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureMinRange"/>
    <DUL:hasDataValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">-50</DUL:hasDataValue>
</owl:NamedIndividual>
```

Figure 3.7 En excerpt of the description of temperature sensor with SSN



```
<!-- http://purl.oclc.org/NET/ssnx/ssn#TemperatureObservation20 -->

<owl:NamedIndividual rdf:about="http://purl.oclc.org/NET/ssnx/ssn#TemperatureObservation20">
    <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureObservation"/>
    <featureOfInterest rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#BedsorePatient430"/>
    <observationResult rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureOutput20"/>
    <observedBy rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureSensor1"/>
    <observedProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#Temperature"/>
</owl:NamedIndividual>



<!-- http://purl.oclc.org/NET/ssnx/ssn#TemperatureOutput20 -->

<owl:NamedIndividual rdf:about="http://purl.oclc.org/NET/ssnx/ssn#TemperatureOutput20">
    <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureOutput"/>
    <hasValue rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureValue"/>
</owl:NamedIndividual>



<!-- http://purl.oclc.org/NET/ssnx/ssn#TemperatureValue -->

<owl:NamedIndividual rdf:about="http://purl.oclc.org/NET/ssnx/ssn#TemperatureValue">
    <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#TemperatureValue"/>
    <DUL:hasDataValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int">25</DUL:hasDataValue>
    <DUL:isClassifiedBy rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#Celsius"/>
</owl:NamedIndividual>
```

Figure 3.8 SSN description of a possible observation generated by a Semantic temperature sensor

3.3.1.3 Virtual Semantic sensor

We presented before, that an IoT application joins the network with the aim to have the needed information from sensors to apply some reasoning techniques. it is also able to provide new analyzed information. Hence, these applications can also be seen as sensors since they are detecting sensor data as stimulus and generating new information as observation. In order to define this type of sensors we will rely on the virtualization paradigm.

Virtual objects have been discussed in some research studies in IoT. In [19] authors present the virtual sensor as any entity (device, object or people) capable of observing properties around them. In [178], authors surveyed virtual objects in the IoT world and they introduce the virtual object as a major solution for IoT platforms problems today. Moreover, authors in [179] show that virtualization in IoT world allows to reduce the resource consumption and to enhance the service discovery. It also promotes the scalability, heterogeneity, QoS Management, Mobility, devices discovery, security and privacy. It has a fundamental role to bridge the gap between the physical and the virtual world.

We define, a **virtual semantic sensor** as a software component with a major objective to translate the needs and requirements of a specific application into higher semantic description. Based on this description, it provides a personalized semantic model of a semantic sensor. Unlike the semantic sensor, the input of a virtual sensor is not a physical stimulus, it is rather the semantic observation sent by a semantic sensor. Semantic observation can be also seen as a semantic stimulus. The output of virtual semantic sensor is a new personalized description of semantic sensor observations.

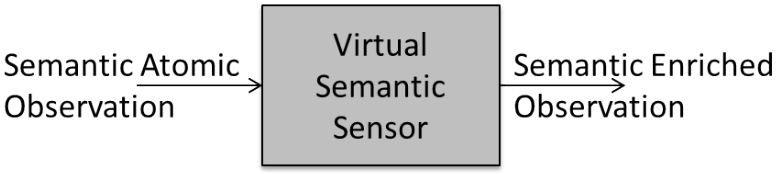


Figure 3.9 Virtual Semantic Sensor

For each defined semantic sensor several virtual semantic sensor can be created depending on the needs of the IoT system. In that case, the same sensor is used by several applications and for different contexts. Moreover, it allows to use the same sensor in different frequencies and for a defined period of time.

To give an example, a pedometer sensor is deployed around a wrist of an elderly person and a semantic description of this sensor is generated. The property of this sensor is defined as general one like "steps" to fit all applications. For the same semantic sensor, two virtual semantic sensors with two different properties can be defined in AAL context.

First, an application defines a virtual semantic sensor which has the objective to monitor person's activity status each 1 hour. Hence, it receives semantic observations, analyzes person's steps and generates his status like "the person is on wheelchair" or "the person walks occasionally". In this case, the property of this sensor is defined as "activity". Another virtual semantic sensor with the aim to analyze the mobility of the person every two hours can be created for the same semantic pedometer. Same, it receives semantic observations related to the steps and infer the mobility such as "mobility is slightly limited". The property of the sensor in this case is "mobility".

Hence, the same observations sent by the semantic pedometer are generated by virtual semantic sensors and contextualized for "activity" or for "mobility".

Another example is manifested by the possibility to create a virtual semantic sensor that convert Celsius values to Fahrenheit based on the single semantic representation of a temperature sensor. For the same sensor, another virtual semantic sensor can be created to detect temperature values and trigger an alarm when the value reaches a predefined threshold.

The virtual semantic sensor presents a high level of semantic enrichment, it can be seen as a specific description and contextualization of a semantic sensor. A virtual semantic sensor with a smartness level 2 is a data consumer and data provider for specific-domain IoT application as we introduced in section 2.3.

Observations sent from virtual semantic sensors are then received by IoT applications which will use reasoning techniques to be more efficient and smart. Hence, IoT applications which employs advanced reasoning techniques can generate high level of knowledge in their observations. In our approach, these applications are considered as cognitive sensors that we describe in the following section.

3.3.1.4 Cognitive sensor

The term cognition is not new in computer science, in the past it has been a reference for adaptation, learning methods [180], intelligent networks [181] and making decisions about the future [182]. This term has been integrated today in IoT world in order to make it smarter and more efficient. The authors in [183] introduce five ‘pillar’ technologies of IoT today: sensor and actuator networks, identification and tracking technologies, enhanced communication protocols, distributed intelligence, and cognitive technologies. Cognitive IoT has been also discussed by IBM [184] and defined as a second generation of Intenet of things technologies. Harriet Green- GM says in the IBM document that "In the era of Cognitive IoT, no machine is an island ". Moreover, the term cognition has been referred to the ability to be aware of the environment, to be able to learn from the past actions, and to use it to make future decisions that benefit the network in [185].

As powerful as sensor networks are today, they are not up to generate knowledge information that is more useful then simple sensor data, for health diagnosis for example or for pollution monitoring in a lake. Cognition in IoT means adding intelligence and smartness to the network and spread new information performed by cognitive mechanisms. Cognitive mechanisms include, intelligent collection, analysis, reasoning and fusion of basic and data generated by several sensors and then, extracting knowledge from this information. Indeed, machine learning and semantic technologies are predominant techniques in IoT world for intelligent data processing [183]. By infusing intelligence into systems and processes, IoT applications will be able to not only do things more efficiently, but to improve the possibility to discover, anticipate and monitor several diseases in the healthcare domain.

Cognitive sensor is any application or processor that receives as input semantic data from different semantic sensors or virtual semantic sensors like OWL temperature observations and OWL humidity observations. It is capable of processing these information by reasoning tools, machine learning, semantic or any decision making techniques. It is then able to communicate the new information to other system components. As the new information reveals high level of knowledge based on aggregation of smart semantic information, we consider that a cognitive sensor presents a high level of smartness.

As an IoT system component, cognitive sensor can be seen as a specific type of the virtual semantic sensor since it is both a data consumer and data provider. The difference consists of the level of information and knowledge that each one can provide. Moreover, the cognitive sensor processes data from different data sources.

Concretely, a medical application represented by a bedsore risk monitoring application represents a cognitive sensor. This sensor processes semantic data of temperature, humidity activity and moisture sensors. It then uses Braden scale assessment to calculate the risk factor of an elderly person and processes the risk as its sensor output. This sensor output is the result of a high level of knowledge like " a person has a high risk of occurrence of bedsore ". This use case is fully detailed in the section 3.6.

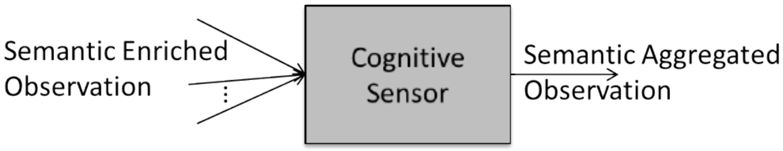


Figure 3.10 Cognitive Sensor

Virtual semantic sensor and cognitive sensor are therefore, domain-specific components are necessary to define expert knowledge and integrate it into the IoT system. They make the IoT network more adaptable, intelligent and useful for a specific application domain like the healthcare domain. However, they can not be described using the SSN ontology. Virtual semantic sensors and cognitive sensors will be described by the CoSSN ontology developed and described in section. 3.4.

We have defined a sensor hierarchy based on intelligence and smartness, it is also necessary to define a data model related to sensors: What kind of messages do they exchange and what are the main information that is exchanged between these smart things. In the next section we define a smart observation model with its knowledge and intelligence characteristics.

3.3.2 Definition of Smart Observation Model

In IoT world, the main function of a sensor is to detect environmental and sensed measurements and send a readable message to the network. To refer to the sensed measurement, we will use the term observation as the result of sensed property and sensor data to the network. Observation concept is defined in SSN ontology and it allows to add more information than just a value or a sensorOutput. As there are different data sources (physical sensor, semantic sensor, virtual semantic sensor and cognitive sensor) the sent observation does not involve the same level of knowledge.

There are different ways of exposing data in the internet through different formats, under different data models. When an application gathers data from a set of particular sensors and use some ad-hoc data model, it creates an isolated data set with very few possibilities of re-use or integration. The use of ontologies and OWL format tackles the different representation of data. However, sensors observations can have different precision and semantic granularity. A classification of the different generated observations is described in the proposed Observation Model in Figure 3.11.

Physical sensor, the basic data provider, generates raw data namely sensor output in ontological language. In fact these data are not shared over the network but only used by the semantic sensor in order to generate semantic sensor description. In our approach, three possible observations can be transmitted over the IoT network depending on the kind of sensor under use. First, "observation" is a first layer of semantic description of physical sensor output. It contains the value, the property, the time of the sensed property and the location. The virtual semantic sensor output is defined as a "contextualized observation" where the property fields is dynamic and changes from an application to application. Finally, the observations resulting from the aggregation of different observations or resulting from some reasoning techniques is classified as "Aggregated Observation". It is generated by cognitive sensor and described as cognitive knowledge since it refers to analyzed information.

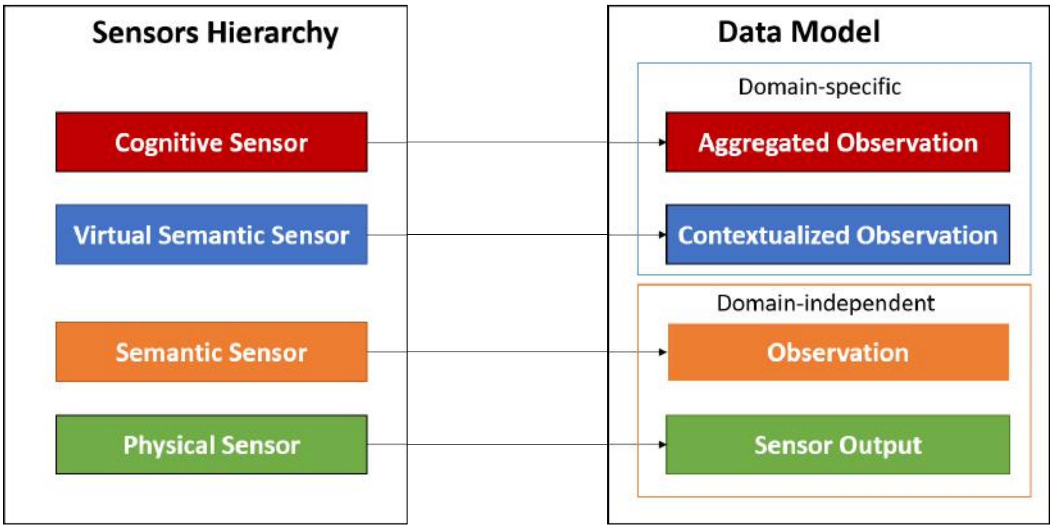


Figure 3.11 Relation between sensors hierarchy and data Model

3.4 Design of CoSSN Ontology for IoT-based domain description

SSN Ontology is a suitable solution for describing physical sensors and creating semantic description of deployed sensor network. However, in IoT environment actuators and applications that virtually act as sensors cannot be described through SSN. Our main objective of semantic description is to allow the description of any entity including software components that detect some properties and provide new information that is sent as observation. For that, we designed the CoSSN ontology as a global and generic ontology to describe IoT applications. In this section, we describe our methodology in designing CoSSN and then we describe its concepts.

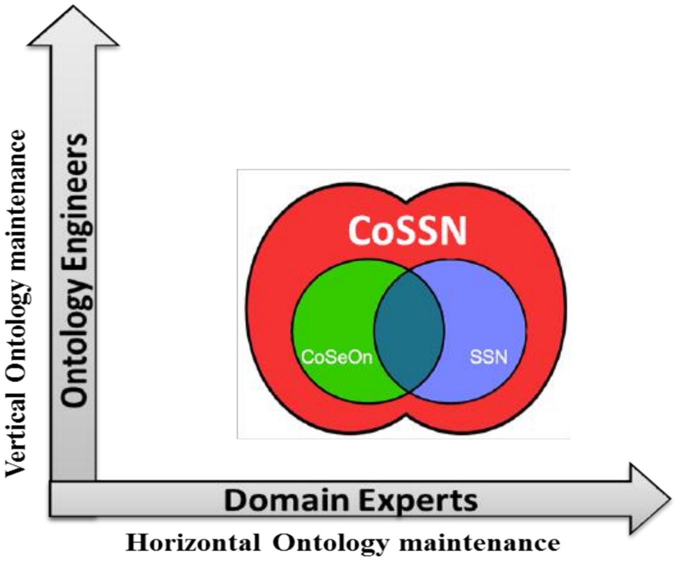

Vertical Ontology maintenance
Ontology Engineers
CoSSN
CoSeOn
SSN
Domain Experts
Horizontal Ontology maintenance

Figure 3.12 horizontal and Vertical Ontology engineering.

3.4.1 Design Methodology

In order to build an ontology-based semantic model, we consider the two dimensions of ontology engineering: vertical and horizontal. These concepts have been introduced by Siorpaes and Hepp in [186]. The vertical ontology engineering refers to extending an ontology by axioms and adding new general concepts. The horizontal engineering refers to extend an ontology by concepts and properties but not in the level of detailed axiomatization. While the horizontal engineering is a typical task for domain experts, the vertical ontology engineering is a task for ontology engineers (see Figure. 3.12). Therefore, SSN is our nexus ontology and we extend it from vertical and horizontal engineering perspectives.

From a vertical ontology engineering perspective, we designed CoSeOn ontology to represent information and facts that inferred from actual sensor data. It describes also experts knowledge and the cognitive process aiming at interpreting such information. CoSeOn extends SSN vertically by adding new axioms mainly related to "Virtual Semantic Sensor" and "Cognitive Sensor" description.

From horizontal ontology engineering perspective, some related concepts should be defined for the healthcare domain. These concepts describe mainly the different types of health sensors (ambient and on-body). Moreover, we added spatio-temporal concepts to describe the location of these sensors. As SSN is a generic ontology and presents some limitation in presenting time and space, it was cumbersome to define these new concepts.

The resulting ontology Cognitive Semantic Sensor Network ontology (CoSSN) is a generic ontology that can be extended with application domain concepts such as those related to healthcare. It includes conceptual entities from SSN, from CoSeOn and from their intersection. Fig. 3.13 depicts the relationships between the ontologies under use.

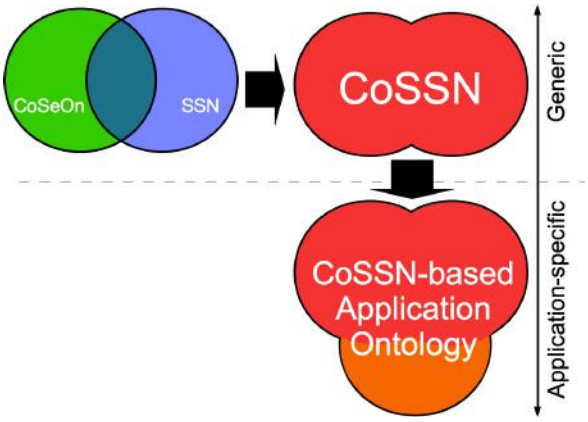


Figure 3.13 Relationships between ontologies used for cognitive sensing of activities in ambient assisted living

3.4.2 CoSeOn Ontology: Cognitive Sensor Ontology

CoSeOn was created to describe sensors which aim at detecting events and facts that need to be interpreted by a cognitive process. The ontology is generic so that it can be extended with application-domain concepts such as actions and activities of interest for the healthcare sector. With respect to SSN, this new ontology puts the emphasis on the cognitive sensing method which relies on experts knowledge, on the virtual entities to be sensed and on how they are related. In this section, we describe the new concepts defined in CoSeOn and we present the Ontology Design Pattern used to implement CoSeOn.

3.4.2.1 CoSeOn Ontology Design Pattern

Similarly to SSN, CoSeOn was defined following the stimulus-sensor-observation ontology design pattern (see Figure. 3.2). It is a reusable solution to the recurrent problem of modeling how sensors observe properties of features of interest by detecting cognitive stimuli, i.e. modifications in the physical world related to these properties, and how cognitive sensors transform stimuli in results to cognitive sensor output. Figure 3.14 presents how this design pattern is implemented in CoSeOn and shows the differences in blue with respect to SSN. We define four main concepts that constitute the ODP of CoSeOn: Cognitive sensor, cognitive stimulus, cognitive sensing and cognitive sensor output.

- **cognitive stimulus** is a change in an environment that can promote a better awareness and comprehension of surroundings. It derives from one or more physical stimuli (e.g., temperature rise, lowering pressure) detected by physical sensors (see Fig. 3.15). Examples of cognitive stimuli in ADL context are actions (e.g., opening the wardrobe, opening the fridge) and activities (e.g., person is preparing his meal, person is dressing up) performed by humans.

  It should be noted that a **cognitive stimulus** is semantically connected and aligned with the **stimulus** concept of SSN through the **derivingFrom** relationship (see Fig. 3.15). This is a key element allowing the integration of CoSeOn, SSN and of the corresponding conceptual layers: the one related to the cognitive process and expert knowledge for CoSeOn and the other related to the actual physical sensing for SSN.

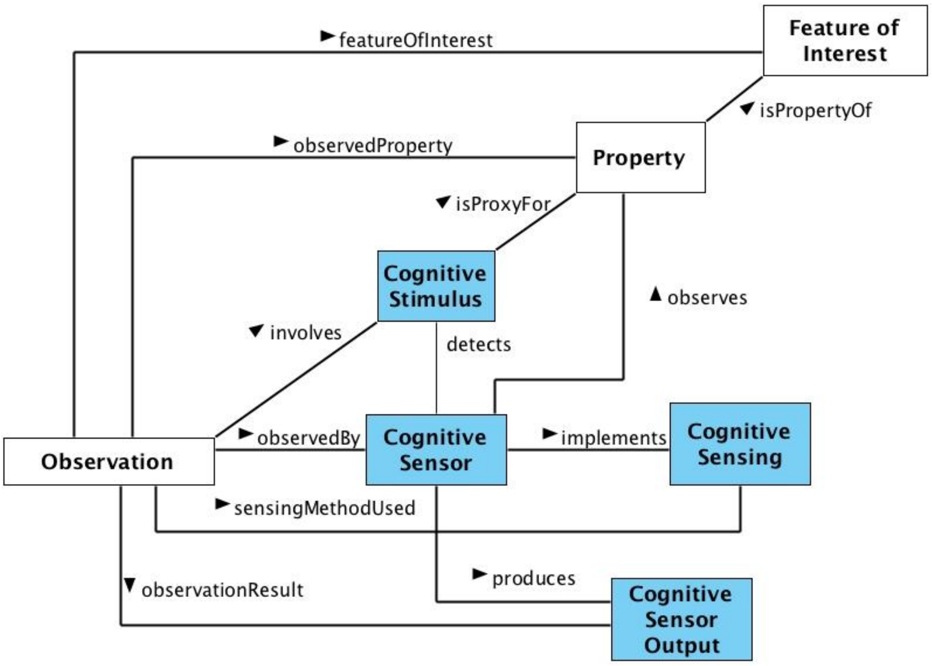


Figure 3.14 Implementation of the Stimulus-Sensor-Observation design pattern in the CoSeOn ontology.

- **Cognitive sensing** is a computer-based process that emulates human-mind perception and interpretation of one or more cognitive stimuli. It leverages on semantically-enriched data and a reasoning mechanism (e.g., defined by a logical rule, as a SPARQL query, or an algorithm) which can be used, for instance, to detect diseases risk factors, actions and activities of a person.

- **Cognitive sensor** is a virtual sensor that performs observation by transforming an incoming stimulus into a sensor output as described in the section 3.3.1.4. In CoSeOn, this component is described as follows: A *Cognitive Sensor* detects some *Cognitive Stimulus*, implements some *Cognitive Sensing* and produces some *Cognitive Sensor Output*. It observes some *Property* and it is observed by some *Observations*.

- **Cognitive sensor output** represents the precessed data generated by a cognitive sensor and the informative outcome of the cognitive sensing process (e.g., the detected action or activity). In CoSeOn ontology, it is related to the Observation concept by the *ObservationResult* relationship and to the Cognitive Sensor by the *produces* relationship. It should be notices also that in IoT, sensor Output includes the most important sharable information since it includes data values.

As stated above, CoSeOn includes several concepts that can be used to describe domain concepts such as human **actions** and **activities** to be detected by appropriate cognitive sensors. By way of illustration:

- An **action** is a cognitive stimulus that represents an atomic step within an activity (see Fig. 3.15). It refers to the inferred information arising from actual data generated by one or more sensors. These sensors are deployed in a specific place in the person's environment. For example, ”opening the wardrobe” is an action detected by a button sensor placed in the wardrobe.
- An **activity** is a cognitive stimulus that represents a set of one or more correlated actions or activities which are detected in a smart place at a specified time or duration. For example, ”person is preparing his meal” is an activity which results from the correlation of ”opening the fridge”, ”turn ON the oven”, ”washing vegetables” and ”preparing the table” actions.

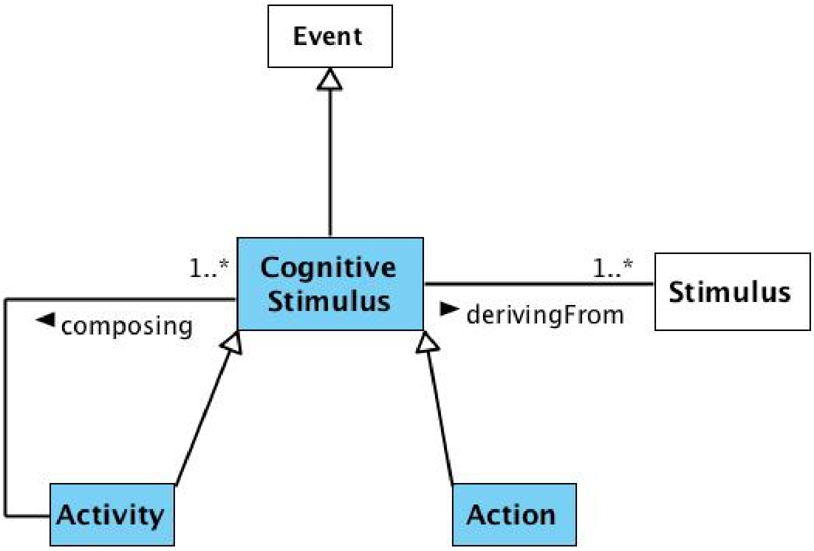


Figure 3.15 Concepts specializing "Cognitive Stimulus", relationship between "Cognitive Stimulus" and "Stimulus", and "composing" relationship between "Cognitive Stimulus" and "Activity"

3.4.3 SSN extension for Healthcare Concepts Description

As previously stated, SSN is a domain-independent ontology and it should be extended vertically in a way to present domain requirements. So, to integrate all aspects of the healthcare domain, we have to describe the following points:

- The different kinds of physical sensors: In AAL and medical domain, applications rely mainly on data provided by wearable and ambient sensors. For that, we believe that it is important to classify physical sensors and to integrate this classification into SSN. This classification allows to describe precisely characteristics of sensors with their specific location in the home environment.

- Spatio-temporal deployment of medical sensors: by classifying wearable and ambient sensors types, make it obvious to define two different types of location to locate each type. Ambient devices are basically deployed in the apartment whereas wearable devices are deployed on the body of the person. Moreover, a sensor can be deployed in different locations at a different moment. This specification is also integrated in the extension of SSN in order to enhance the precision of monitoring applications.

3.4.3.1 Wearable and Ambient Sensors

In order to monitor people we have mainly two types of sensors: wearable sensors including inertial sensors and Ambient sensors. To monitor health measurements, wearable devices such as bracelets and watches are capable of sending information related to heart rate, physical activity, sleep information, and more other measurements. In AAL context, ambient devices are the main data sources as the person is monitored based on its location at home. In such environments, we seek information like "open door", "watching TV", "Toileting" gathered by ambient sensors like contact, light and presence sensors.

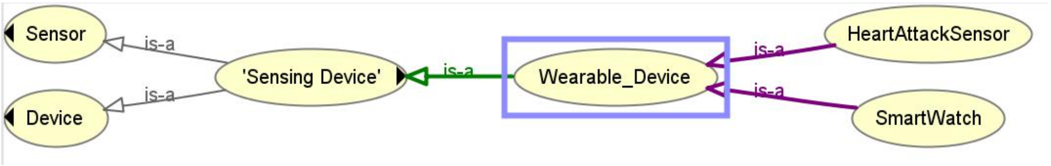


Figure 3.16 Wearable sensors concepts added to SSN ontology

SSN is then extended with two subclasses of sensing device class (ssn:Sensingdevice): "Wearable Devices" as presented in Figure. 3.16 and "Ambient Devices" as presented in Figure. 3.17. Basically, all sensors types can be created such as smart watch, temperature, water, button and so on. Based on these sensors types, deploying new sensors in an AAL environment requires the creation of an instance of the corresponding sensor's type. Hence, a sensor can be easily added and integrated to the network using the predefined classes of sensors.

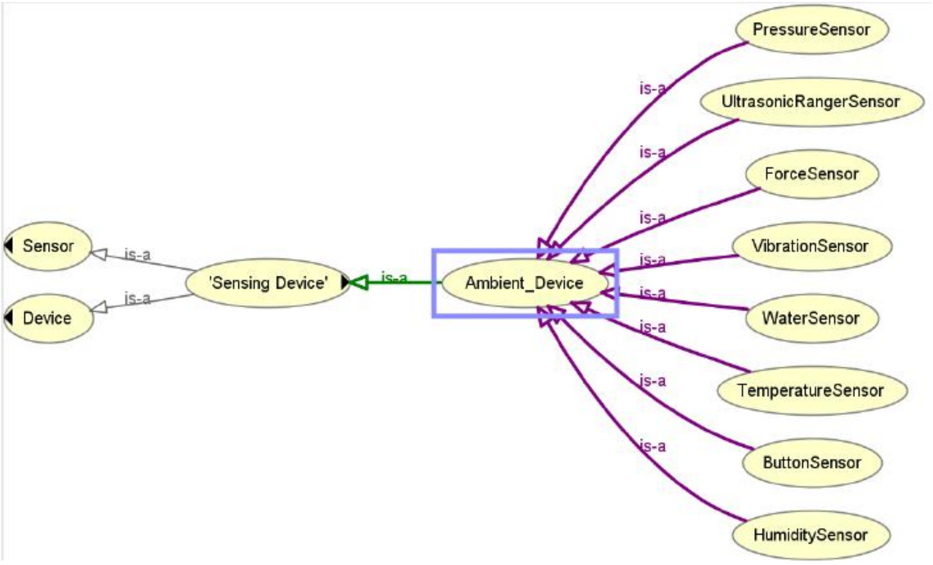


Figure 3.17 Ambient sensors concepts added to SSN ontology

3.4.3.2 Spatio-Temporel Concepts

SSN presents some concepts promoting the description of spatio temporal region related to a physical sensor by the concept *ssn:SpatioTemporalRegion*. The Spatio temporal concept has two components: *ssn:SpaceRegion* and *ssn:TimeInterval*. Although, in SSN the location of each sensor (ambient or wearable) is described by the expression *DUL:hasLocation* some *ssn:SpatioTemporalRegion*. But, as we have defined two kinds of sensors, an enhancement should be added to the ontology for a more precise description. For instance, in the context of an hospital, retirement homes or personal homes, geographic parameters like latitude and longitude can't be pertinent and do not fit the characteristics of localization of the person. For that, an internal description of "zones" of the apartment or the person's location is required. Moreover, for the wearable devices, we identify its location based on the "body part" concepts.

Hence, we identify two new concepts to add to SSN in order to describe the location of sensors "zone" and "body part":

Zone: A space region has a specific dimensional space that is used to localize an Entity. We define a "zone" as any place and location that has a specific dimensional space around the person. It is specified according to the granularity required by an application. For example, a room is a zone, a bed is a zone and a wardrobe is a zone as depicted in Figure. 3.18 where we suggest that the person is at home. Based on this definition, we can describe the location of an ambient sensors in such way: a "CoSSN:AmbientDevice" DUL:hasLocation some *ssn:SpatioTemporalRegion* which is constituted by some "*CoSSN:zone*" and some *ssn:timeInterval*. In such case, we can define a specific location for an ambient device at a certain time interval and change it at another time. For illustration, an example of a temperature sensor deployed in a patient bed for a week period is presented in Figure 3.19.

In this example, we describe a temperature sensor defined as *CoSSN#TemperatureSensor1* that observes *CoSSN#Temperature* property. This sensor has been deployed in the bed32 from Tuesday09-08-2017 until Tuesday16-08-2017 as it is described by the *DUL:hasConstituent* concept.

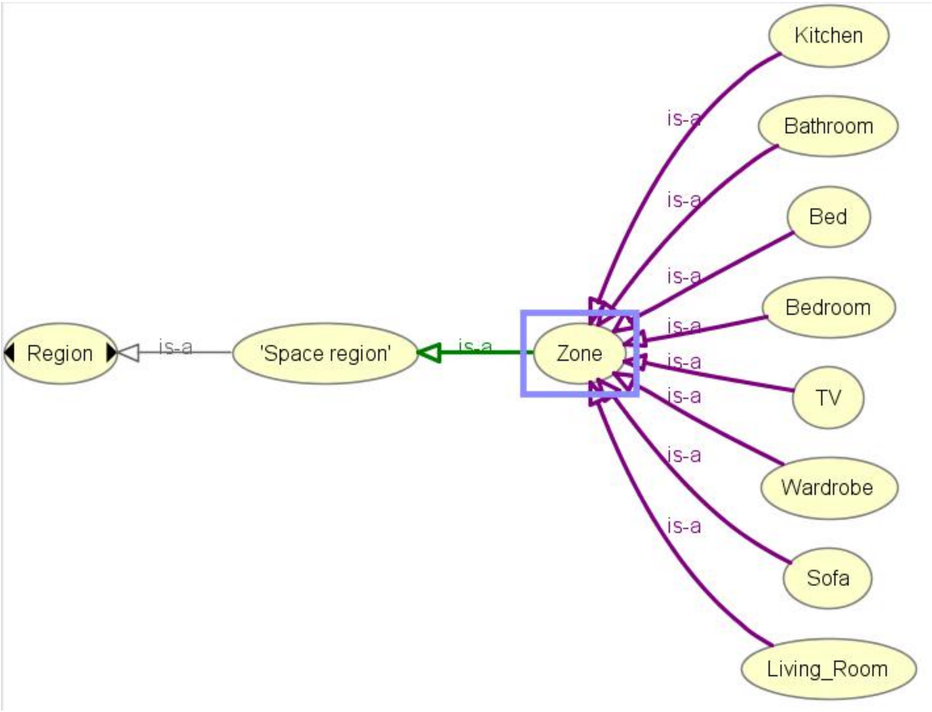
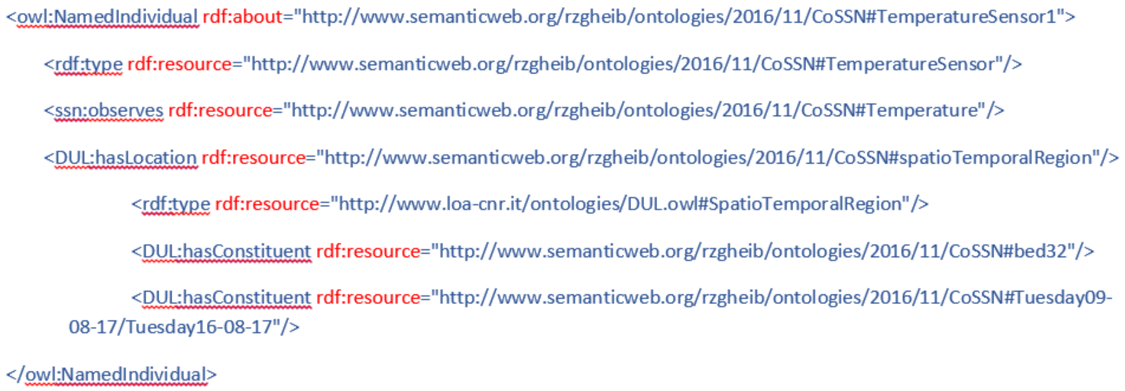


Figure 3.18 Defintition of the zone concept for ambient sensor localization



```
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#TemperatureSensor1">

    <rdf:type rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#TemperatureSensor"/>

    <ssn:observes rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#Temperature"/>

    <DUL:hasLocation rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#spatioTemporalRegion"/>

            <rdf:type rdf:resource="http://www.loa-cnr.it/ontologies/DUL.owl#SpatioTemporalRegion"/>

            <DUL:hasConstituent rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#bed32"/>

            <DUL:hasConstituent rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#Tuesday09-
      08-17/Tuesday16-08-17"/>

</owl:NamedIndividual>
```

Figure 3.19 Description example of a temperature sensor deployed in patient bed32 for one week

Body Part: Wearable devices are usually deployed on a person. Based on the definition of the location of on-body sensors in the Opportunity project [2], we added the different body part under the Platform concept as depicted in Figure. 3.20. It is defined in SSN ontology as an Entity to which other Entities can be attached such as Sensors and other Platforms. Similarly to ambient sensors, a "wearable_device" DUL:hasLocation some "ssn:SpatioTemporalRegion" but it is also characterized by the "ssn:onPaltorm" object property in SSN and applied on some "CoSSN:Body_part".

Discussion

We have presented in this section the design of our semantic approach CoSSN which is an extension of the SSN ontology. We designed the CoSeOn ontology to breach the virtual processing description gap in SSN. In fact, at the time of writing, a new draft of SSN is being under study (see https://www.w3.org/TR/vocab-ssn/). In this new draft, some propositions are made to describe virtual sensors including our own views on the subject. To address this issue, we extended the SSN ontology following a design methodology in order to handle the new virtualization paradigm of semantic sensors.

Moreover, ontologies provide the ability to discover devices described in a specific environment by querying data and information in OWL format. In the following section, we present some reasoning rules that can be applied to CoSSN.

[2] http://www.opportunity-project.eu/challengeDataset

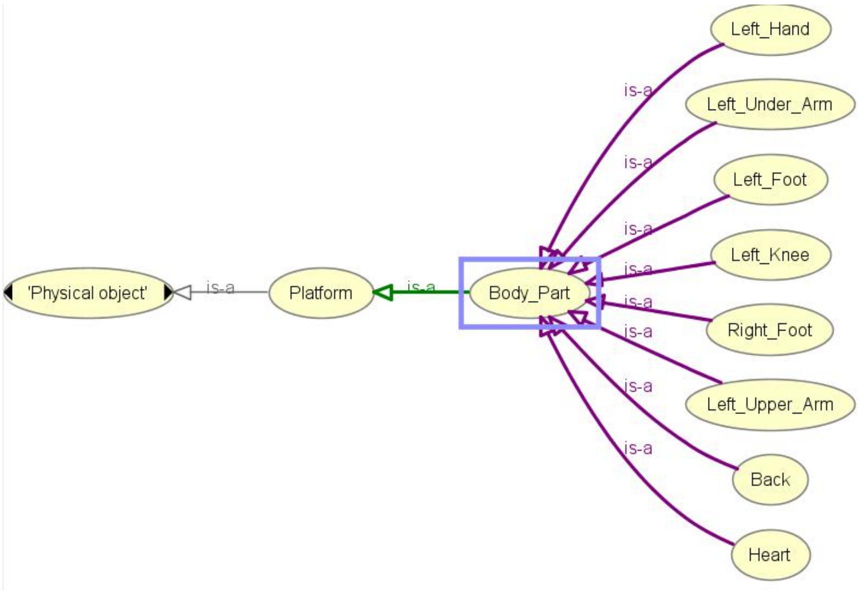


Figure 3.20 Defintition of the body part concept for wearable sensor localization

3.5 Querying Semantic Sensor Networks

One of the key ingredients of the Ubiquitous Web and IoT world is the management of data streams, which may be obtained from a wide range of data sources, from social networks to environmental sensors. The use of ontology as a semantic description technique has led to acquire RDF to express these data streams, possibly according to well-established vocabularies, such as the W3C Semantic Sensor Network Ontology. In fact, semantic data are usually stored as RDF triples in a database called triplestore or RDF store.

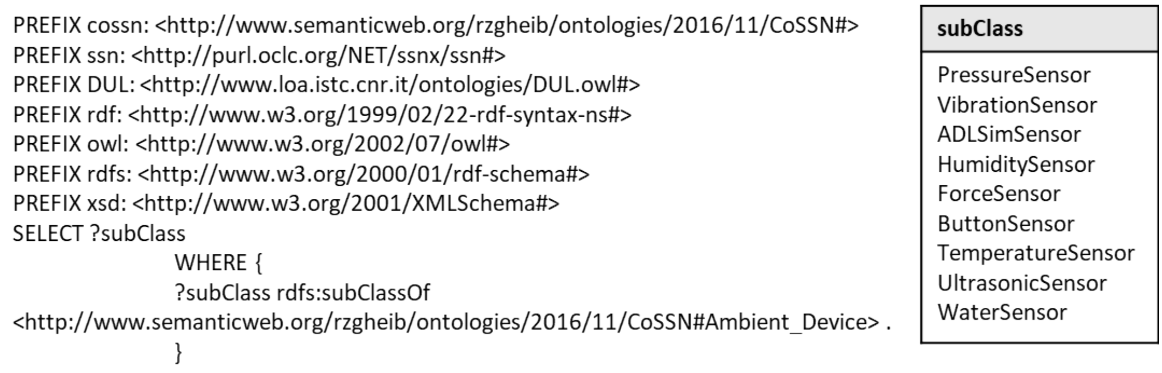
As presented in the previous section, a semantically rich sensor network will provide spatial, temporal and thematic information essential for discovering and analyzing sensor data. The growth of available information in the network requires new efficient data access methods by domain experts in charge of making decisions.

Triplestore or RDF store is a purpose-built database for the storage and retrieval of triples through semantic queries. The ultimate query techniques for ontologies is the SPARQL [74], a recursive acronym for SPARQL Protocol and RDF Query Language. SPARQL provides and efficient technique that support an interactive data exploration and provides the domain experts the ability to access and analyze available data sources without involving IT experts. Indeed, querying semantic sensors is necessary and it has several main duties in our approach:

- Discovery of existing sensors in the network;
- Detection of existing properties of the different sensors and observations;
- Selection of the appropriate sensors according to expert knowledge needs;
- Reasoning mechanism to define logical rule or an algorithm which can be used to solve domain-dependent issues such as to detect bedsore risk or activities for ADL monitoring.

SPARQL Query Language

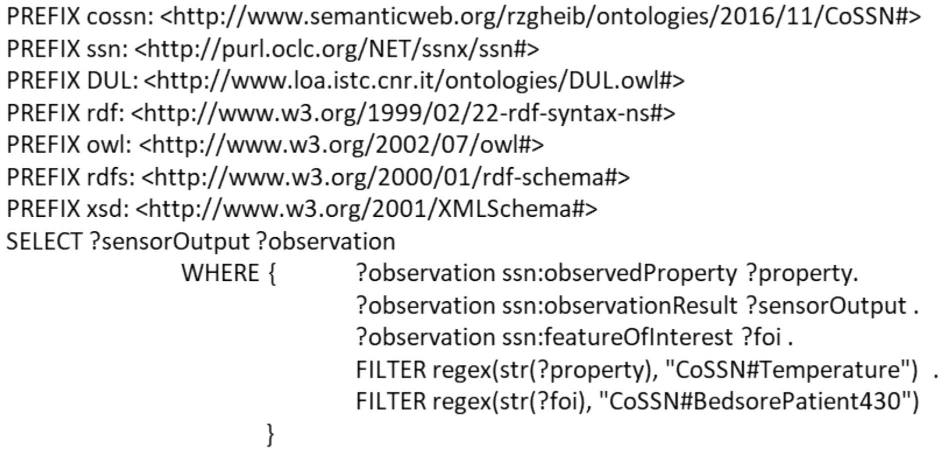
Today, the SPARQL query language is an important technique to apply queries and rules on ontologies. Figure. 3.21 presents a SPARQL query that can be applied to CoSSN ontology in order to retrieve all ambient sensors defined in the IoT system. Figure. 3.22 shows a SPARQL query that retrieves temperature observations and values from a temperature sensor deployed for the patient "BedsorePatient403".



```
PREFIX cossn: <http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX DUL: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subClass
                WHERE {
                ?subClass rdfs:subClassOf
<http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#Ambient_Device> .
                }
```

subClass
PressureSensor
VibrationSensor
ADLSimSensor
HumiditySensor
ForceSensor
ButtonSensor
TemperatureSensor
UltrasonicSensor
WaterSensor

Figure 3.21 SPARQL query retrieving Ambient sensors created in the ontology. This query generates all types of sensors that have been created in an IoT system.



```
PREFIX cossn: <http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX DUL: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?sensorOutput ?observation
            WHERE {      ?observation ssn:observedProperty ?property.
                         ?observation ssn:observationResult ?sensorOutput .
                         ?observation ssn:featureOfInterest ?foi .
                         FILTER regex(str(?property), "CoSSN#Temperature")  .
                         FILTER regex(str(?foi), "CoSSN#BedsorePatient430")
                  }
```

Figure 3.22 SPARQL query retrieving sensor output and observation for a Temperature property with feature of interest the patient identified by "BedsorePatient403".

3.6 Bedsore Risk Detection Illustrative Case Study

3.6.1 The Bedsore Disease

A bedsore (also called pressure ulcer) [187] is a localized injury resulting from prolonged pressure on the skin. It plagues persons who stay in bed or wheelchair for a long period of time. More than just a wound, the bedsore pathology can cause physical and moral suffering, a limitation of autonomy and an often underestimated repercussion on the quality of life of the patient. It can have important consequences, leading to long-term hospitalization. The patient is at risk of surgery and even of death.

In retirement homes and hospitals, bedsores pathology is a main concern since prevention techniques in hospitals and retirement homes today are still traditional. The medical/nursing staff spends a considerable amount of time checking (usually every 30 minutes) the status of their patients and their changes in body position. Based on a specific assessment scale, they evaluate the status of the patient and fill a paper sheet each time. It should be noticed that development of a bedsore is considered a serious fault from the healthcare team.

In the research area, preventing and treating bedsores has become a global challenge for today's healthcare researchers. Some studies have found that patients with mobility impairment, in coma or in long surgical procedure [188] develop bedsore easily.

Furthermore, the average length of a stay is 3 times higher for patients with bedsore according to the European Pressure Ulcer Advisory Panel (EPUAP). From the other side, in the context of home care for dependent elderly people, the frequent checking for bedsore risks are not possible anymore with the increasing number of elderly worldwide.

Therefore, society needs a way to predict the early formation of pressure and needs to have an accurate bedsore detection system based on real time sensors measurements. Intelligent systems such as bed detection system [189] and Intelli-Sense Bed [190] has been proposed as solutions to monitor bedsore. However, these solutions are based on 1 or 2 criterias like temperature and not on a medical assessment.

In light of the challenges and problems arising from the traditional monitoring of bedsore, we decided to propose an automated assessment of bedsores based on IoT paradigm.

3.6.2 Automatic assessment for detecting the risk of bedsores

Several assessment scales [191] have been proposed in the literature in order to quantify the risk of bedsores among which are the Norton scale [192], the Waterlow scale [193] and the Braden scale[194]. According to domain experts in nursing school IFSI, the school with whom we have collaborated , the Braden scale is the most used method in clinical settings especially in France. It results from a simple calculation based on 6 risk factors: sensitivity, mobility, nutrition, activity, moisture and friction. In the current medical practice, the risk factors are assessed by nurses or medical aids through clinical examination or interviews with the patient. In fact, for each patient, a nurse creates manually a dashboard collecting information and assesses each risk factor with a value in the scale [1..4]. This value corresponds to the intensity of this factor for the patient. A global score is then calculated that determines the risk of developing a pressure ulcer or bedsore as depicted in Figure. 3.23. The lower the score, the higher the risk of bedsore occurrence.

In order to illustrate our CoSSN ontology in the context of healthcare applications, we propose to deploy smart sensors in the patient's bed or chair in order to monitor several of the Braden scale's criteria. For instance, the friction criteria is monitored by a pressure sensor. The moisture criteria is detected by monitoring the wetness level of the bed sheet thanks to a moisture sensor. Finally activity and mobility criterias can be monitored by a pedometer.

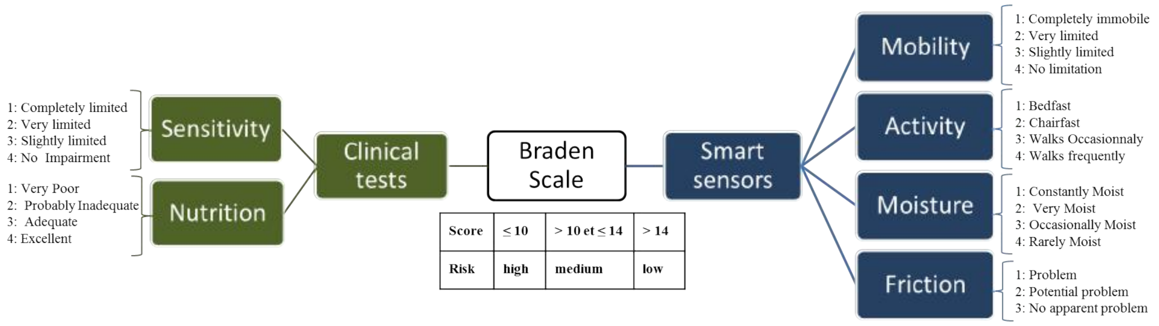


Figure 3.23 Braden Scale

Braden score attributes	Score factor			
	1	2	3	4
Moisture value	> 950	750 - 950	300 - 750	<300
Activity value (Pedometer)	0	0	10 - 20 steps	> 20 steps
Mobility value (Pedometer)	0	5 - 10	10 - 20 steps	> 20 steps
Pressure value	> 50	30 - 50	10 - 30	0 - 10

Table 3.1 Intelligent sensors used for Braden scale calculation and the threshold values for each one

In order to automate the assessment of Braden scale, a study on the physical sensors values has been conducted. Furthermore, it was necessary to study the braden scale factors to define the threshold sensors values and finally to assign for each factor score a range of values. Based on that, a summary of the four factors moisture, activity, mobility and pressure values is presented in table 3.1, for each attribute a score is assigned corresponding to sensor range values.

3.6.3 Bedsore Risk Detection Modeled with CoSSN

Based on the previous study of the possible sensors that can be deployed to asses the risk of bedsore, a bedsore application can be created and modeled with CoSSN. Designing the bedsore application involves three steps (see Figure.3.24) that an IT developer must follow:

1. **Description of Physical Sensors:** An IT developer must define the used physical sensors in CoSSN with their characteristics in term of measurement capabilities, deployment and properties. When an IoT system is already installed

at home for example and sensors types already exist, the IT developer can skip this step and creates directly instances of the required sensors. The result of this step is the generation of the semantic sensors (see Figure. 3.7). In this case study, three semantic sensors are generated: SemanticMoisture which observes Moisture property, SemanticPressure which observes Pressure property and SemanticPedometer which observes Steps property.

2. **Definition of Virtual Semantic Sensors (VSS):** Braden scale consists of defining scores for each factors in order to compute the overall bedsore risk score. VSS are developed to analyze the received observations and to perform new contextualized observations that indicate the score of a factor for a specific patient. Four VSS can be developed, the first *VirtualSemanticPressure* that observes FrictionScore property and generates its Braden score. Second, the *VirtualSemanticMoisture* that observes MoistureScore and generated its Braden score.

   Finally, two VSS are created for the same SemanticPedometer sensor. In fact, based on the steps generated by the same pedometer, activity and mobility factors can be detected in the context of bedsore detection. The Semanticpedometer defined in CoSSN has a property "CoSSN:steps". virtualSemanticPedometer1 is defined in CoSSN with the property "CoSSN:activity", it requests the steps information, analyzes it and communicates the score of activity factor. And, virtualSemanticPedometer2 is defined in CoSSN with the property "CoSSN:mobility" requests the steps information, analyzes it and communicates the score of mobility factor. An excerpt of the defined VSS is presented in Figure.3.26.

   Hence, the steps number sent from the pedometer for the property "steps" is analyzed by different components at the same time and sent by a virtual sensor under different property (steps, activity, mobility).

3. **Definition of the Cognitive Sensor:** The bedsore application is defined as a cognitive sensor in CoSSN with a property "CoSSN:bedsore". It receives the contextualized observations from the virtual semantic sensors, applies the calculation of the Braden scale score and outputs a new aggregated observation. The new observation contains a high level of knowledge that can be expressed as "there is a high risk of bedsore for patient 3".

As previously stated, physical sensors and semantic sensors are domain-independent. An IT developer can then bypass the first two steps when an IoT healthcare system is already deployed and the required sensors are already defined for other applications. Regarding to virtual semantic sensors, the IT developer can check if another application already generates the needed information. Otherwise, he should create new virtual semantic sensor as a subclass of the VirtualSensor class in CoSSN. The cognitive sensor must always be defined for new arrival application.

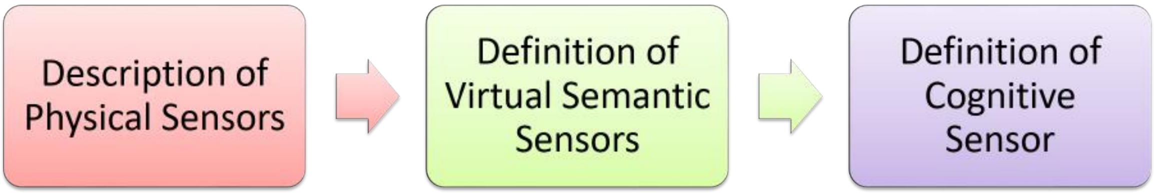

Description of Physical Sensors
Definition of Virtual Semantic Sensors
Definition of Cognitive Sensor

Figure 3.24 IT programmer steps to describe an IoT-based bedsore application in CoSSN

3.6.4 Scenario of a Person with a High Risk of Bedsore Modeled with CoSSN

In order to ease the presentation of the case study, we will present the scenario of an elder identified by "Patient 3". This patient suffers from chronic disease and he has been plagued in his bed for a long time. In his bed, two sensors are deployed (moisture, pressure) and he is equipped with a pedometer. For instance, when the moisture value is 500, pressure value is 30, the activity is 0, and the mobility between 5 and 10, then the person is at high risk of bedsore occurrence.

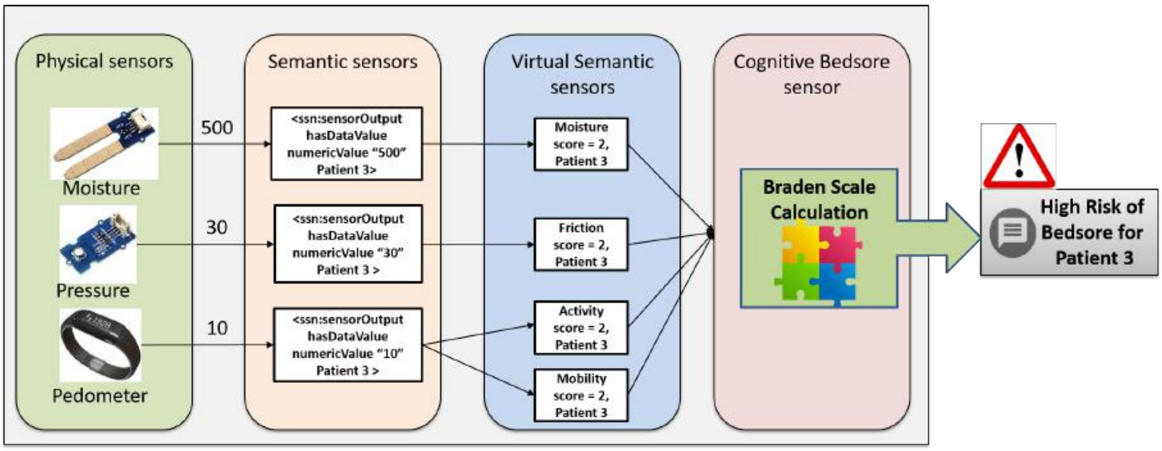
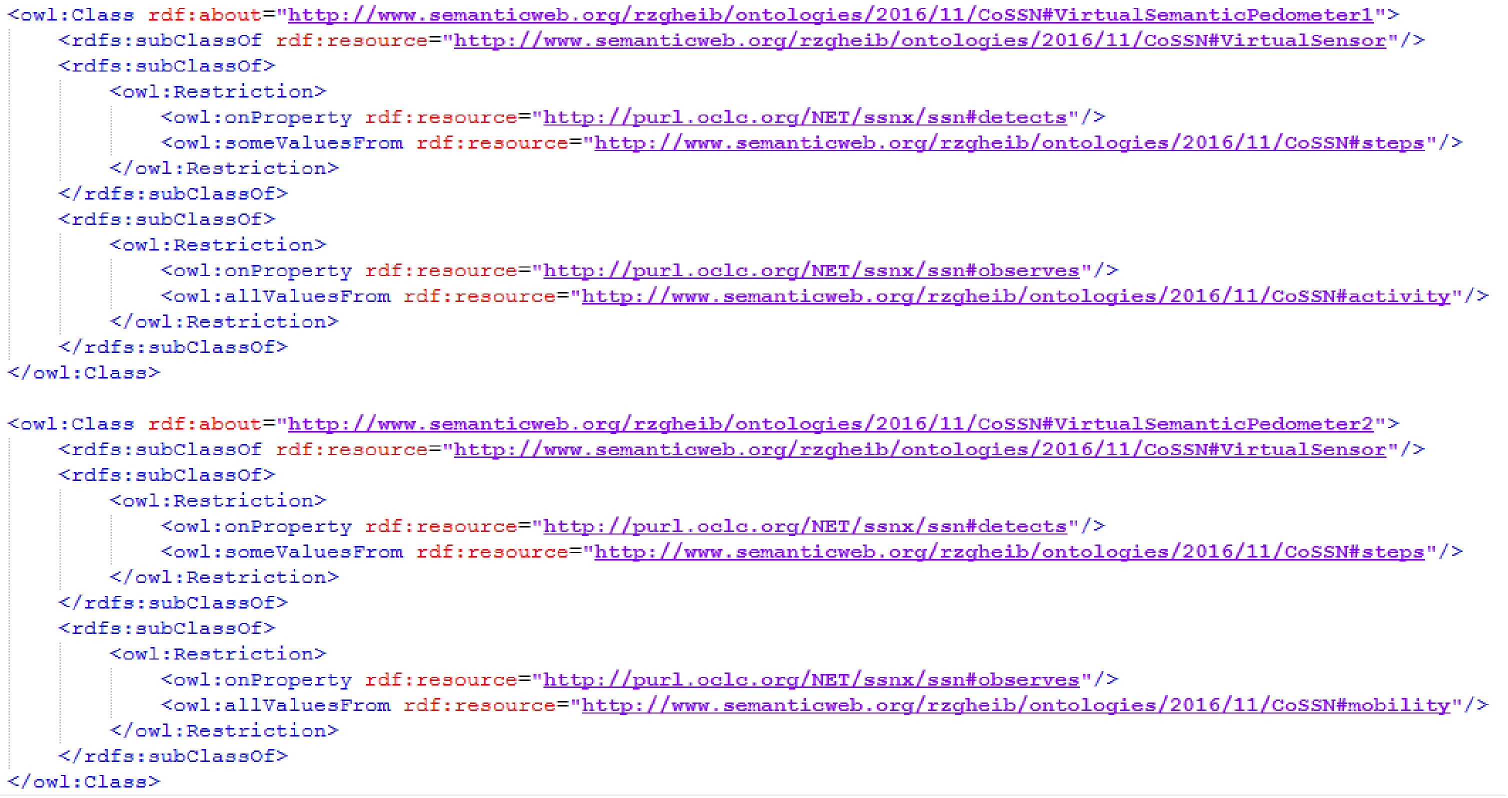

Physical sensors
Moisture
Pressure
Pedometer
500
30
10
Semantic sensors
<ssn:sensorOutput hasDataValue numericValue "500" Patient 3>
<ssn:sensorOutput hasDataValue numericValue "30" Patient 3 >
<ssn:sensorOutput hasDataValue numericValue "10" Patient 3 >
Virtual Semantic sensors
Moisture score = 2, Patient 3
Friction score = 2, Patient 3
Activity score = 2, Patient 3
Mobility score = 2, Patient 3
Cognitive Bedsore sensor
Braden Scale Calculation
High Risk of Bedsore for Patient 3

Figure 3.25 Modeling of Bedsore Risk Detection

We present the different components of the scenario as we presented in the previous section in Figure. 3.25. In this example, the moisture sensor is sending the value 500, pressure sensor 30 and the pedometer 10. Semantic sensors have encapsulated the numeric value in a rich RDF message. The virtual semantic sensor defined for moisture sensor has analyzed the value "500" as a high value in the bed of patient 3 and has assigned the score "2". Similarly to other virtual semantic sensors, the score of pressure factor is "2", activity is "2" and mobility is "2".

Finally, the defined bedsore risk application receives the observations from the virtual semantic sensors. It applies the calculation and outputs a new aggregated observation containing a high level of knowledge that can be expressed as "there is a high risk of bedsore for patient 3".



```
<owl:Class rdf:about="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#VirtualSemanticPedometer1">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#VirtualSensor"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#detects"/>
            <owl:someValuesFrom rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#steps"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#observes"/>
            <owl:allValuesFrom rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#activity"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#VirtualSemanticPedometer2">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#VirtualSensor"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#detects"/>
            <owl:someValuesFrom rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#steps"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#observes"/>
            <owl:allValuesFrom rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#mobility"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 3.26 An excerpt of virtualSemanticPedometer1 and virtualSemanticPedometer2 description in CoSSN

3.7 Conclusion of the chapter

In this chapter, we have presented the knowledge description and data description in IoT environments. We put a light on ontologies for semantic data description for its historical role in data discovery and linking; device discovery and selection; and querying sensor data, tasking and reasoning performance. We provided a detailed description of SSN with its main ontology design pattern (ODP) concepts highlighting its limitation.

We defined a smart sensor hierarchy reflecting the different types of sensors and virtual sensors that can be found in an IoT system. The main three data providers are the **semantic sensor** which wraps the typical physical sensor and presents a domain-independant description. Contextualizing and cognitive processing of sensor data are represented through the **virtual semantic sensor** and **the cognitive sensor**. We presented also the corresponding observation model reflecting the generated information from each sensor.

We detailed our proposal, the CoSSN ontology, for a full semantic description of an IoT system. It integrates two main ontologies SSN and CoSeOn. CoSSN relies on SSN to describe physical sensors and on the proposed CoSeOn ontology to describe the knowledge and cognitive processing of semantic sensor data. This approach provides a full description of sensors, shared data. It also provides the ability to infer high level of information that reflects experts knowledge. CoSSN enhances the semantic interoperability through an IoT system. It also promotes the collaboration between all system components by exchanging OWL messages.

We illustrated our approach with a bedsore risk detection case study in the healthcare domain. We showed how an IoT application can use our solution to define sensors and to describe their data on a semantic way. Moreover, we showed how our semantic proposal enhances the reasoning by applying calculation and rules easily such as the Braden scale calculation for the bedsore case study.

To sum up, we presented the semantic components that can interact in an IoT system as well as the type of messages that can be exchanged. However, to define a full IoT architecture, a gateway and a communication paradigm should be deployed for easy and flexible sharing of information as described in section 2.3. In the next chapter we present the communication model of the proposal as well as the integration of CoSSN in this model. Hence, we will present how CoSSN is used as the semantic basis for the SeMoM architecture described in the next chapter.

Chapter 4

SeMoM: A Semantic Message Oriented Middleware for IoT Applications

"Design is not just what it looks like and feels like? Design is how it works."

— *Steve Jobs*

Contents

4.1 Introduction

We presented in the previous section CoSSN, a part of our contribution which aims at providing a full description of IoT components in intelligent environments. However, our main objective is to share semantic information through a flexible and collaborative architecture. In order to design an IoT architecture, a communication model must be defined. From a communication point of view, solutions providing strong coupling between data providers and consumers make data proprietary and the integration for new applications more complex.

The loose coupling of components as we described in section 2.3 allows for the addition of new data providers without any modification from the data consumer perspective. One of its main functionalities also is the ability to change/replace a data provider, leaving the rest of the system untouched. In order to allow a loosely coupled communication, we propose a message oriented middleware (MoM) architecture with a publish/subscribe paradigm. From the other side, adding semantic description to a middleware architecture has been presented in the semantic web of things paradigm which has been introduced as a novel paradigm to generate knowledge in an IoT system. Based on that, we propose in this chapter a semantic web of things architecture, SeMoM, for semantic IoT applications.

In this chapter, we provide in section 4.2 an overview of existing IoT architectures that offer both semantic and communication features. We present then in section 4.3 the semantic Web of Things paradigm that support our proposal. Based on that, we present the principles of our proposed architecture. Afterwards, we put a light on the MoM concepts and we present our SeMoM architecture for IoT applications. Finally, we illustrate our approach in section 4.4 with an healthcare case study application: the bedsore risk detection before concluding the chapter in section 4.5.

4.2 Existing Middleware Approaches in IoT

Our objective in this thesis is to integrate semantic description in an IoT middleware architecture which is a core component in many IoT platforms. An IoT platform is a suite of components that enable developers to deploy IoT based applications including management of devices and sensors. Middleware platforms are gaining more and more importance due to their major role in simplifying the development of new IoT applications as well as abstracting the complexities of the network communications or even hardware considerations. This abstraction allows the application developer to focus all his effort on the task to be solved. But, the need for semantic description in middleware platforms, adds a new complexity regarding the developer since new semantic libraries should be added for ontologies management. For that, there is a limited use of semantic techniques in most of IoT middleware platforms.

Moreover in healthcare, the programmer should have a minimum knowledge of the domain in order to add the specific health concepts to the ontology and the application. In some work, a collaboration between experts and programmer is planned in order to develop a domain-specific application. For example, diabetes monitoring application or heart attacks monitoring applications require reasoning on domain-related information rather than just simple smart sensors signals. A suitable platform for such applications including health management does not exist in the literature. However, there are many IoT platforms proposed for sensors and data management in an IoT environment. In the following, we present an overview of semantic IoT middleware approaches in the literature as well as middleware based platforms for IoT and we position our work regarding their proposals.

4.2.1 Overview on Existing Semantic Middleware Approaches

Existing approaches have considered semantic description in proposing middleware architecture for IoT applications. SOA and MoM approaches are prominent IoT architectures where semantic enhancements have been added. For instance, authors in [195] propose an application layer solution for interoperability for IoT aiming to resolve the interoperability issue between different kinds of protocols (Bluetooth and UPnP). The key idea is to use device semantics provided by existing specifications and dynamically wrap them in their SOA middleware into semantic services.

openAAL [196] is an open source middleware for AAL applications, it relies on SOA architecture as a communication paradigm and on ontology description enabling service discovery. It defines a framework on top of the OSGi specification to facilitate integration and communication among services, including the context manager, procedural manager, and composer.

SENSEI [197] is a SOA middleware for IoT. It includes a context model, context services, actuation tasks, and dynamic service composition of services. The main component of this middleware is the resource layer, which stands between the application layer and communication services layer. Resources in SENSEI use ontologies for their semantic modeling.

LinkSmart, SOA middleware [198] relies on a semantic model-driven architecture and enables the use of devices as services. The semantic description of devices is based on ontologies.

ubiSOAP [199] is a lightweight service oriented middleware that offers resource management and network level interoperability by supporting heterogeneous networking devices and technologies. Dynamic composition and instantiation of new services are facilitated by the semantically rich models and XML descriptions of sensors, actuators, and processing elements.

KASOM [200] is a knowledge-Aware Service Oriented Middleware proposal for pervasive environments. Its architecture consists in offering services through registration, discovery, composition, and orchestration of services. Most of these services are established on complex reasoning mechanisms and protocols based on a contextual model, which represents a semantic description of low- and high-level resources of the WSN. Real-life implementations in hospital and health management show its potential in terms of response time, efficiency, and reliability. However, KASOM does not provide dynamic service composition in mobile and resource constrained IoT infrastructures because of predefined service composition rules provided by in-network agents.

CHOReOS [201] enables large scale choreographies or compositions of adaptable, QoS-aware, and heterogeneous services in IoT. It provides a scalable probabilistic thing-based service registration and discovery to address scalability, interoperability, mobility, and adaptability. It uses ontologies for semantic enrichment over the architecture. The semantic thing based service compositions are transparently and automatically executable with no involvement from end-users, which is highly desirable in IoT, especially in M2M communications.

SMArc [202], is an acronym for Semantic Middleware Architecturethe and a SOA Middleware Architecture. It focuses on smart city energy management for smart grid environments. A light ontology and data representation under a specific format have been used in the implementation in order to guarantee that the interchanged data uses a representation format which will be common in the system, and therefore information will be easier to extract.

OM2M [203] is an advanced semantic middleware based on SOA architecture. It is a Machine-to-Machine service based on autonomic computing and semantic annotation to provide an inter-operable system to connect billions of devices but they do not consider real-time analysis and full loosely coupled architecture. Authors proposed the IoT Ontology (IoT-O) [150] for the autonomic management of M2M systems. They extended SSN with four main modules: Acting module to describe actuators, Lifecycle module to model state machines, Service module to describe services and finally the Energy module to represent the power consumption for appliances. However, this ontology does not consider the description of virtual sensors and cognitive processing of information.

Another example is the European project OpenIoT [19] that has developed an open-source middleware platform providing a "cloud-of-things". OpenIoT aims to propose on demand access to cloud-based IoT services for internet-connected objects. Trying to use sensing as a service, OpenIoT architecture embeds the CUPUS middleware as a cloud-Based publish/subscribe processing engine and relies on SSN for sensors description. The observed data is stored as linked data and processed based on SPARQL queries which are continuously executed as data arrive. It can be viewed as a federation of several middlewares interconnected with each other targeting applications for smart cities or campus and agriculture domain. OpenIoT could have been a good candidate to target IoT health applications, but its complexity due to the variety of middlewares under use can be a major drawback for programmer.

Semantic MoM-based proposals have not been widely discussed in the literature regarding SOA-based proposals. However, new solutions have considered MoM and semantic description in their research work. Namely, EnTimid, xAAL and SITRUS middleware proposals.

EnTimid [204] is a MOM middleware for smart home monitoring. It relies on component model for sensors description but it does not address messages or domain description in this contribution.

xAAL [17] is another example of MOM middleware, it has been designed in the context of PRECIOUS [1] European project for AAL. Sensors schema model has been proposed in this work for sensors description and system components exchange Json messages between them.

Bispo et al. propose the SITRUS middleware in [205]. Beside a MOM communication model, authors rely on ontology as semantic description processing module whose purpose is to generate a semantic database that provides the basis to decide whether a WSN node needs to be reconfigured or not.

4.2.2 Overview on Existing Semantic Middleware Platforms

The previously mentioned approaches have been presented in the literature as semantic middleware solutions however, other middleware solutions have been introduced through IoT platforms. In [1] [206] [207], authors present a large review of 38 contemporary IoT platforms. They described each platform and classified them based on seven characteristics: support of heterogeneous devices, type of the platform, architecture's design, proprietary or open source, support of REST, data access control and service discovery.

As presented in Table 4.1, the second column shows if the platform support heterogeneous devices or otherwise what kind of sensors it handles. We can see that most of the described projects support heterogeneous devices. Some solutions like Everyware needs a gateway to manage devices and there are solutions that support only one kind of sensors like Fosstrack that supports only RFID sensors.

We also deduce that in most cases, the platforms are provisioned from a cloud, either in a form of a Platform-as-a-Service (PaaS) or a Software-as-a-Service (SaaS). This type of platforms provides storage facilities, devices management, device connectivity, backup mechanisms or online support.

The table also includes information about the openness of the platforms. Open-source platforms are considered more promising compared with the proprietary alternatives because they are expected to enable the faster integration of new IoT solutions across the application domains. Among the platforms presented in the table, only 11

[1] http://www.thepreciousproject.eu/

are open-source.

Moreover, it may be deduced from the table that only a few platforms do not have a REST API. This demonstrates that the current IoT services will tend to adopt the web of things paradigm [208]. We also deduce that only a few platforms have integrated some type of service discovery mechanisms. A comprehensive survey on discovery protocols for M2M communications can be found in [209].

The presented solutions have not been studied from a semantic perspective. However, our main objective in this thesis was to add semantics to IoT platforms. For that aim, we added the last column in the table to show if a platform integrates semantic description or not. We also present the used data format when semantic description is used. As it is depicted in the table, few platforms have integrated semantic descriptions in their solution. JSON and XML are predominant used formats in both open and proprietary platforms.

Table 4.1 Comparative table (Recap) of IoT Platforms [1]

Platforms	Support of heterogeneous devices	Type	Architecture	Open source	REST	Data access control	Service discovery	**Data description**
AirVantage	Needs gateway	M2M PaaS	Cloud-based	Libraries only (Apache v2, MIT and Eclipse v1.0)	Yes	OAuth2	No	**n.a.**
Arkessa	Yes	M2M PaaS	Cloud-based	No	n.a.	Facebook like privacy settings	No	**n.a.**
ARM mbed	Embedded devices	M2M PaaS	Centralized/ Cloud-based	No	CoAP	User's choice	No	**n.a.**
Carriots	Yes	PaaS	Cloud-based	No	Yes	Secured access	No	**JSON, XML**
DeviceCloud	Yes	PaaS	Cloud-based	No	Yes	n.a.	No	**n.a.**
EveryAware	Yes	Server	Centralized	No	Yes	4 levels	No	**JSON, XML**
Everyware	Needs gateway	Paas	Cloud-based	No	Yes	n.a.	No	**JSON, XML**
EvryThng	Yes	M2M PaaS	Centralized	No	Yes	Fine-grained	No	**n.a.**
Exosite	Yes	Paas	Cloud-based	Libraries only (BSD license)	Yes	n.a.	No	**JSON**
Fosstrack	RFID	Server	Centralized	No	No	Locally stored	No	**n.a.**
GroveStreams	No	PaaS	Cloud-based	No	Yes	Role-based	No	**JSON**
Hub-of-All-Things H.A.T	Home devices	PaaS	Decentralized	Yes	Yes	Locally stored	Yes	**n.a.**
IoT-framework	Yes	Server	Centralized	Apache license 2.0	Yes	Locally stored	Yes	**n.a.**
IFTTT	Yes	SaaS	Centralized	No	No	No storage	Limited	**n.a.**
Kahvihub	Yes	Server	Centralized	Apache license 2.0	Yes	Locally stored	Yes	**n.a.**
LinkSmart	Embedded devices	P2P	Decentralized	LGPLv3	No	Locally stored	Yes	**ontologies to discover devices**
MyRobots	Robots	Robots PaaS	Cloud-based	No	Yes	2 levels	No	**JSON, XML**
Niagara	Yes	M2M SaaS	Distributed	No	n.a.	n.a.	n.a.	**n.a.**
Nimbits	Yes	Server	Centralized/ Cloud-based	Apache license 2.0	Yes	3 levels	No	**JSON**
NinjaPlatform	Needs gateway	PaaS	Cloud-based	Open source hard-ware and Operating System	Yes	OAuth2	No	**JSON**
Node-RED	Yes	Server	Centralized	Apache license 2.0	No	User-based privileges	No	**n.a.**
OpenIoT	Yes	Hub	Decentralized	LGPLv3	No	User-based privileges	Yes	**ontologies to discover devices**
OpenMTC	Yes	M2M client/server	Centralized/ Cloud-based	No	Yes	Secured access	No	**n.a.**
OpenRemote	Home devices	Server	Centralized	Affero GNU Public License	Yes	Locally stored	No	**n.a.**
Open.Sen.se	Ethernet enabled	PaaS / SaaS	Cloud-based	No	Yes	2 levels	Limited	**JSON**
realTime.io	Needs gateway	PaaS	Cloud-based	No	Yes	Secured access	No	**JSON**
SensorCloud	No	PaaS	Cloud-based	No	Yes	n.a.	No	**n.a.**
SkySpark	No	SaaS	Centralized/ Cloud-based	No	Yes	n.a.	No	**n.a.**
Swarm	Yes	PaaS	Cloud-based	Client is open source (unknown license)	Yes	n.a.	n.a.	**JSON**
TempoDB	No	PaaS	Cloud-based	No	Yes	Secured access	No	**n.a.**
TerraSwarm	Yes	OS	Decentralized	n.a.	n.a.	n.a.	Yes	**n.a.**
The thing system	Home devices	Server	Centralized	M.I.T.	Yes	User's choice	No	**n.a.**
Thing Broker	Yes	Server	Centralized	Yes	Yes	Locally stored	No	**n.a.**
ThingSpeak	Yes	Server	Centralized/ Cloud-based	GNU GPLv3	Yes	2 levels	Limited	**JSON, XML**
ThingSquare	Embedded devices	Mesh	Cloud-based	Gateway firmware is open source	Yes	No	No	**n.a.**
ThingWorx	Yes	M2M PaaS	Cloud-based	No	Yes	User-based privileges	Yes	**n.a.**
WoTkit	Yes	PaaS	Cloud-based	No	Yes	Secured access	Yes	**n.a.**
Xively	Yes	PaaS	Cloud-based	Libraries are open source (BSD 3-clause), platform is not	Yes	Secured access	Yes	**JSON, XML**

Beside the before mentioned platforms, there are some open source solutions that have been developed in the context of home automation and AAL environments. We have studied these solutions and classified them in the Table 4.1 taking into consideration how the semantic description in each solution is addressed.

- AllJoyn Lambada [210] relies on Bus-D architecture and offers an open source framework for the management of IoT environments. It can be seen as a software solution integrating AllJoyn in the Lambda architecture used for Big Data storage and analytics. This proposal offers a graphical user interface (gui) for devices discovery and management, secure data transport between communication technologies (Wi-Fi, Bluetooth, etc.), interoperability between different OS and notifications interface that sends/receives human readable messages. A simple smart home case study has been tested using this approach. It consists on turning on/off devices at home. Semantic descriptions have not been considered in this approach.

- Kaa [2] is a server-endpoints platform with the aim to create IoT applications including applications for healthcare, agriculture and smart city. It has been promoted as compatible virtually with any type of connected devices, sensors, and gateways. However, Kaa requires the integration of a specific microchip in the hardware of the IoT device which can be problematic for commercial sensors.

- Mango [3] is a modular web-application framework. The main functionalities offered by this proposal are data acquisition, real time data monitoring, high performance database, logic and automation, security, cross platform, graphic dashboard and internal performance monitoring. It is based on a list of middleware and on an application that are compiled into a single http server object. The middleware and applications are written in a functional style, which keeps everything modular. Semantic description is not provided.

- Nimbits [4] is an open source PaaS that can be used for hardware and software applications. It can be downloaded on platforms like raspberry pi, Amazon EC2 and Google App Engine. This solution has many key features including geo and time-stamped data processing, event and alerts triggering and offer a build provided for Google App Engine and Linux Systems but it does not offer semantic features.

[2] https://www.kaaproject.org

[3] https://github.com/paulbellamy/mango

[4] https://www.nimbits.com/

- OpenRemote [211] is a software integration platform for residential and commercial building automation with the ambition to overcome the challenges of integration between many different protocols and solutions available for home automation. It relies on cloud-based design tools to offer user interface design, installation management and configuration. An internet connection is only required for communication to systems outside the own network, or during configuration. Complex IoT applications with decision module programming are not supported in this platform neither are semantic descriptions.

- servIoTicy API [212] is an IoT-as-a-Service Data Management Platform. It provides multi-provider data stream processing capabilities on the cloud, a REST API, data analytics, advanced queries and multi-protocol support in a combination of advanced data-centric technologies. This work is still in early stages.

- OpenIoT [19] is a generic and open source middleware platform funded by the European Union. It offers open access to a wide range of technologies for Internet-connected sensors and other objects exposed as services with the ability to support large-scale deployments. It offers a friendly user interface and allows to link together different IoT devices and semantic web services.

Commercial IoT platforms have been put into light by technology companies such as Amazon and Microsoft. We summarize some of the industrial IoT platforms in this section.

- Perdix [213] is a PaaS and cloud-based IoT platform made for mainstream sectors like aviation. The main three components are *Predix Machine* responsible for collecting data from industrial assets and pushing it to the *Predix cloud* which is a global, secure cloud infrastructure. Predix Services are used by developers to build, test, and run industrial IoT applications.

- IBM Watson [214] for its part, employs speech and visual recognition, analyses the visual content of images and videos to understand their content.

- AWS from Amazon [215] and Azure IoT Suite [216] present successful IoT platforms in industrial domain. In addition to their private philosophy, industrial solutions do not consider semantic description in their proposals.

Table 4.2 Comparative table (Recap)of Semantic Middleware solutions for IoT

Middleware approaches	Architecture Type		Semantic Description			Application
	SOA	MOM	Sensors	Messages	Domain	
Song et Al.	X		X			Sensors as services
OpenAAL	X		X			IT services, Living Lab
Sensei	X		X			IoT applications
LinkSmart	X		X			Diabetes monitoring and therapy
SM4ALL	X		X			AAL (elderly, disabled people)
EnTimid		X				AAL, home automation
ubiSOAP	X		X			ubiquitous networking
KASOM	X		X			ubiquitous environments
CHOReOS	X		X	X		AAL, home automation
SMArc	X		X	X		Smart city, smart grid
universAAL	X					AAL, home automation
SOPRANO	X		X			AAL, home automation
OM2M	X		X	X		IoT applications
xAAL		X	X	X		AAL, home automation
XGSN (OpenIoT)	X		X	X		OpenIoT, smart city, agriculture
SITRUS		X	X	X		automatic reconfiguration of WSN
AllJoyn	X		X			AAL, home automation
Kaa	X		X			AAL (elderly, disabled people)
Mango	X		X			Diabetes monitoring and therapy
OpenRemote	X		X			AAL, home automation
servIoTicy API		X	X	X		AAL, home automation

Discussion

To sum up, a comparative table is presented in 4.2 with respect to the different perspectives of semantic modeling in IoT described in section 2.4: sensors, messages and domain. This table summarizes the open source semantic middleware solutions cited in this section with respect to the two middleware architectures SOA and MOM. Finally, we provide the application domain that each solution has been proposed for.

As depicted in this table, semantic IoT middlewares have been integrated in different domain applications, namely in AAL and home automation, monitoring of elderly and disabled person, smart city, agriculture, and monitoring of vital and health signs. It is noticeable the predominant choice of SOA architecture as a middleware solution. Moreover, when semantic approaches are proposed they are mainly integrated for semantic descriptions of sensors. But there is no full description including sensors, sensor data and domain.

Moreover, semantic description in SOA architectures has been proposed in order to describe services which refer to sensors specifications in IoT systems. For example, XML schema has been proposed in SM4ALL [217] for sensors description and used as a contract for applications service description.

Other semantic description techniques can be also noticed like model driven architecture for devices description in [218] and Linksmart[5]. Lately, ontologies have gained sufficient importance for knowledge and sensors description as it is presented in SOPRANO [57], Sensei [219] OpenIoT, SMArc ,UBIWARE and OM2M. Another important aspect of this comparison study is the lack of messages and domain description in these propositions, we can see that two ontology-based solutions OM2M and XGSN/OpenIoT have considered OWL format for exchanging messages.

From the other side, only few attempts address the semantic topic in their proposals and most of these attempts rely on existing SOA-based solutions. So we end up with more semantic SOA-based then MOM-based architectures for IoT. But as previously mentioned, to provide a loose coupling and information centric solution, SOA-based solutions are not so relevant. Therefore, MOM in the past few years has been a topic of interest for IoT researcher and some semantic MOM solutions have been proposed. Entimid, XAAL and SITRUS are MOM-based middleware solutions that have addressed semantic representation in their solution. However, these solutions

[5] (http://hydramiddleware.eu/news.php

do not offer a full description model for IoT as it is shown in the table.

Regarding the presented semantic approaches, we present in this thesis the semantic middleware SeMoM for IoT healthcare applications. SeMoM is designed to ensure the overall system interoperability by sharing sensors information among distributed components. Our SeMoM middleware offers a full description of IoT system by means of ontology description. Moreover, SeMoM relies on semantic web of things paradigm that we present in the following section.

4.3 SeMoM: a Semantic Web of Things Architecture

The web of things paradigm can be seen as an evolution of the internet of things where all components share their information and collaborate to generate advanced knowledge such as wisdom. Moreover, adding semantic web technologies across this paradigm would enhances the reasoning and the potential of smart things. In this thesis, the principles on which we rely to design our SeMoM architecture adopt the semantic web of things concepts. These concepts are manifested in SeMoM by adopting a Message Oriented Middleware (MOM) enhanced by adding CoSSN ontology concept. MoM with publish/subscribe communication model is used for flexible data acquisition, near real-time processing of sensor data and data sharing across system components. In this section we briefly present the principles of SeMoM, the MoM communication model as an introduction to the overall architecture and the semantic mapping in CoSSN. Then, we specialize it to healthcare for managing patient health.

4.3.1 Principles of the proposed architecture

Creating networks of “smart things” found in the physical world (e.g., with RFID, wireless sensor and actuator networks, embedded devices) on a large scale has become the goal of a variety of recent research activities. However, they are typically locked into unimodal closed systems. Authors in [126] show that unleashing the full potential of these smart things is guaranteed by providing an open access to sensors. Therefore, sensor data can be integrated with data from several information systems to build new applications. In this context, designing an open access and flexible architecture where all IoT applications and sensors can exchange freely their information, refers to the web of things architecture.

Guinard et al. [124] defined the web of things paradigm as an architecture design that makes the real-world data an integral part of the web. The integration of data on the web demands the use of web technologies (e.g., HTML, JavaScript, Ajax, PHP, Ruby) to build applications involving smart things. Moreover, an overview of the web of things has been presented in [220] as the integration of the physical world with the virtual world of the Internet and thus converting data into wisdom as it is shown in Figure. 4.1.

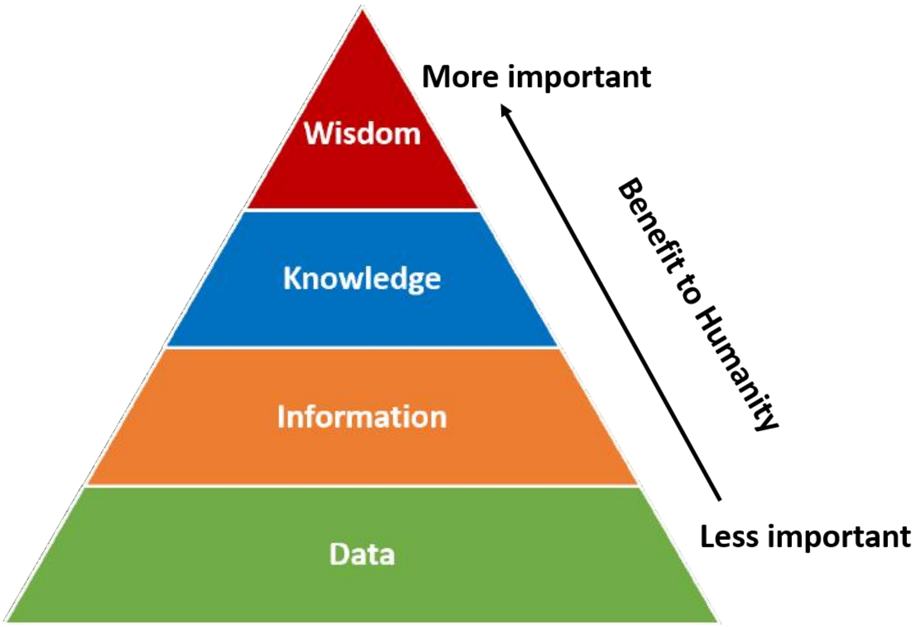


Figure 4.1 Pyramidal representation of web of thing

We rely on the the semantic web of things paradigm in order to design our SeMoM architecture for IoT healthcare applications. SeMoM architecture complies to the principle of "**Independent and weakly coupled software components** driven by **semantic representation** of data and adapted to **domain requirements**" as depicted in Figure. 4.2. This principle is manifested by acquiring the MoM for decoupled and flexible communication and the CoSSN ontology to semantically enrich the exchanged data as well as to describe the domain concepts overall the system.

- **MoM for loose coupling and information pooling:** The loosely coupled communication presents independence between data sources and data providers

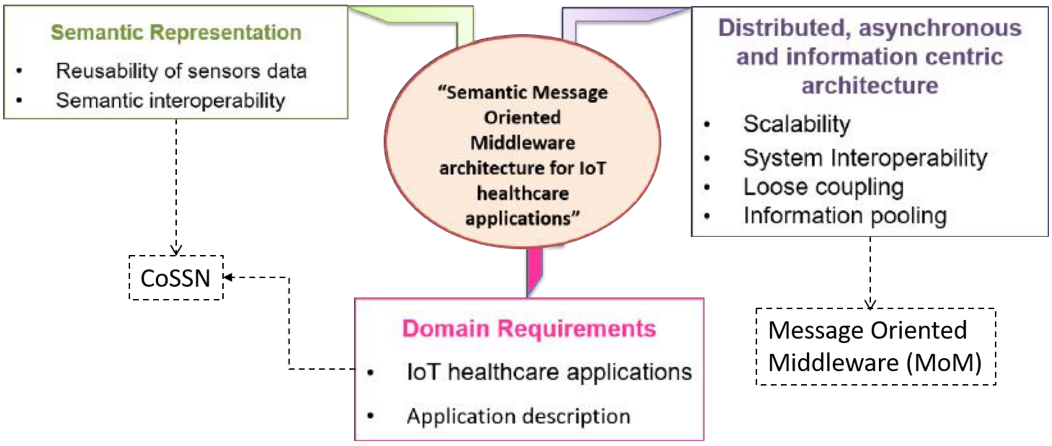

Semantic Representation
Reusability of sensors data
Semantic interoperability
"Semantic Message Oriented Middleware architecture for IoT healthcare applications"
Distributed, asynchronous and information centric architecture
Scalability
System Interoperability
Loose coupling
Information pooling
CoSSN
Domain Requirements
IoT healthcare applications
Application description
Message Oriented Middleware (MoM)

Figure 4.2 Overview of the principles of the proposed SeMoM architecture

which make the communication more flexible. A data provider can then be used by several IoT applications from different applications domains. The replacement of a sensor in such case is transparent to the other components of the system which supports the scalability of devices.

Moreover, it is likely that information from one sensor might be used for several different purposes, by unrelated systems. For instance, movement sensors can be used for measuring overall activity within the home, and also to detect activities of specific interest, e.g. feeding or toilet use. Considering the constant evolution of technological supply, it is also likely that new sensors will be introduced into the running system, to replace, complement or supersede existing ones. A sustainable software architecture must accommodate for obsolescence and improvements of its components, and allow for replacing and introducing new components in a deployed system with minimal impact to the existing applications. For example, in a multi-pathologies case the pooling of physical devices is required in order to decrease the number of redundant deployed sensors around the patient.

Regarding the above stated requirements, we rely on a Message Oriented Middleware (MOM) with a publish/subscribe architecture. We argued in chapter 2 that this architecture is the most adapted approach to use in IoT environments. It focuses on the information itself and supports sending and receiving of messages between distributed systems. This architecture supports the open access to sensors and applications data by supporting the exchange of data between applications.

- **Semantic Representation:** One of the main principles that we follow in our approach is the semantic interoperabitlity. We have discussed in the previous section the important role of ontologies and semantic description in improving the semantic interoperability and enhancing the sharing of formal information across an IoT system. As SeMoM relies on a MoM communication model, the MoM is enhanced with a semantic representation of exchanged data in order to ensure semantic interoperability of the system. The semantic representation includes description of sensor data and metada, sensors observations and enable describing domain concepts in the same ontology which is not proposed by other solutions. Due to the combination of semantic concepts with MoM concepts, new smarter components can be defined involving both semantic and communication features that will allow to generate wisdom to the IoT system.

- **Domain requirements:** By using CoSSN in our architecture, we provide the possibility to describe any application domain. We described in section 3.4 how CoSSN can be used to describe the healthcare domain concepts such as on body sensors, time and location. Moreover, the overall application which enables avoiding the accumulation of redundant sensors when several monitoring applications are used simultaneously, responds to financial concerns and to **patient's desire** to avoid the installation of too many devices in his home.

By combining the three concepts semantic interoperability, loosely coupling and information pooling and health requirements, we can establish a new semantic MOM architecture principle that introduces an "Independent, weakly coupled software components driven by semantic data" for IoT healthcare applications.

4.3.2 Message Oriented Middleware for loose coupled communication

MOM architecture with its publish/subscribe mode promotes low coupling between software components because the source of a specific event (e.g. a sensor) is not mandated to know where, how or for what purpose this data will be processed. Conversely, some specific information (e.g. sleep disorders) can be detected through different means (e.g. pressure sensors under the mattress, or movement detectors). For instance, a system interested in sleep disorder information does not need to be aware of the means used to infer it.

A main objective of this architecture is to provide a scalable and loosely coupled system ensuring interoperability of software components. These components can be viewed as data providers or **Publishers** and data consumers or **Subscribers**. All components communicate through a communication bus, the **broker**. The publisher and receiver exchange messages for a specific field of interest called **topic**.

- **Publisher**: A publisher publishes (sends) information with a specific topic on the middleware. Sensors are the most obvious data providers (publishers) in IoT and in the healthcare domain. These sensors can either be disseminated in the house (ambient sensor) or carried by the inhabitant of the smart house such as for instance body sensors and can rely on wireless or wired communications.

- **Subscriber**: A subscriber registers to one or several specific topics and then consumes every messages labeled with the corresponding topic. It then uses the consumed raw data to perform complex operations. In the healthcare domain, monitoring applications are the main subscribers. For instance, a monitoring application running in the pad of a nurse is an ultimate subscriber to patients status to receive alert notifications about a specific patient.

- **Broker:** In the MOM architecture publishers and subscribers communicate through the mediation of a broker, thus creating a system of loosely-coupled components. With this kind of solution, it is then possible to switch from a provider to another without notifying the consumers/providers as long as the produced data remains available. The addition or the removal of a component is transparent for other components as long as the information is still produced. It provides a strong flexibility and enforces the capacity of the system to evolve.

More specifically, Figure 4.3 presents a sequence diagram showing the different MoM components and the different messages exchanged between them. For instance, the scenario includes one publisher, connected to the broker, which is sending messages for a specific property "topicX". Subscriber1 and Subscriber2 connect to the broker by subscribing to the topic "topicX". The two subscribers then, receive messages (message2 and message3) related to the "topicX" once they subscribe to the broker. The messages sent before a subscription (message1) will not be sent again.

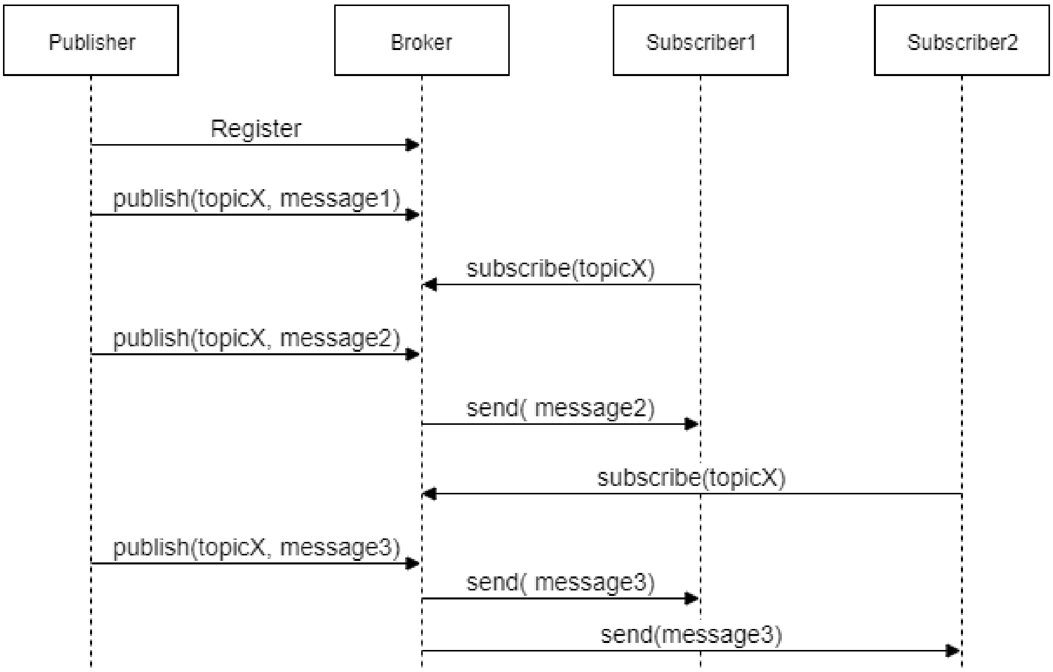


Figure 4.3 Sequence diagram presenting the differ net components of a publish/subscribe architecture

The main two components in publish/subscribe architecture are then the publisher and the subscriber. They are sufficient for many monitoring applications with the aim to visualize sensor data and create statistics. However, in IoT systems where collaboration between domains is necessary, applications remain publishers and subscribers at the same time in order to generate high level of knowledge and publish it to the broker. In order to describe the double role of subscribing and publishing in SeMoM architecture, we will use the term transformers defined in [221].

- **Transformer**: Transformation components can be split in two main categories (see Figure 4.4):

  Simple transformers use the raw data provided by a data producer (i.e publisher) to compute new data to send on the bus. This component will be in charge of publishing knowledge on the bus. For example, a simple transformer subscribes to temperature topic in order to publish an enhanced information such as the action to perform.

  Advanced transformers use raw data as well as data produced by simple transformers and any data producer in the network. Advanced transformers subscribe to several topics and can use and combine different data to deduce new information on the patient or wisdom in web of things terms. For example, temperature data correlated with humidity and activity sensor data related to a patient can be used to detect the braden scale score for bedsore assessment based on the case study presented in section 3.6. This score can then be used to detect if there is risk for the monitored patient.

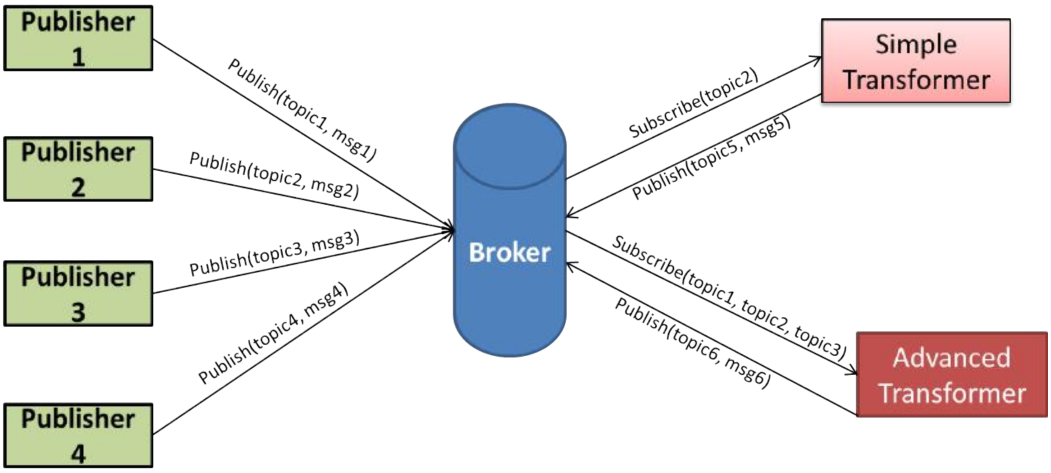


Figure 4.4 Simple and Advanced Transformer in Publish Subscribe architecture

4.3.3 Overview of the SeMoM Architecture

We have defined different components in CoSSN and MoM. In this section, we show how the semantic components have been integrated in the MoM communication model. In CoSSN, we defined semantic components related to the level of knowledge they are performing: semantic sensor, virtual semantic sensor and cognitive sensor. From MoM perspective: publisher, subscriber and transformer are represented as communication components in the aforementioned section.

In SeMoM architecture, we combine the different concepts of CoSSN and MoM in smart communicated components enable to share data through an IoT network. These components refer to four main components: Semantic Publisher (data source), Semantic subscriber (data consumer) Semantic Transformer and the broker. Moreover, we have identified the different possible sensors observations that can be generated by the several components. Sensors observations can be classified into data, information, knowledge and wisdom concepts defined in web of things paradigm.

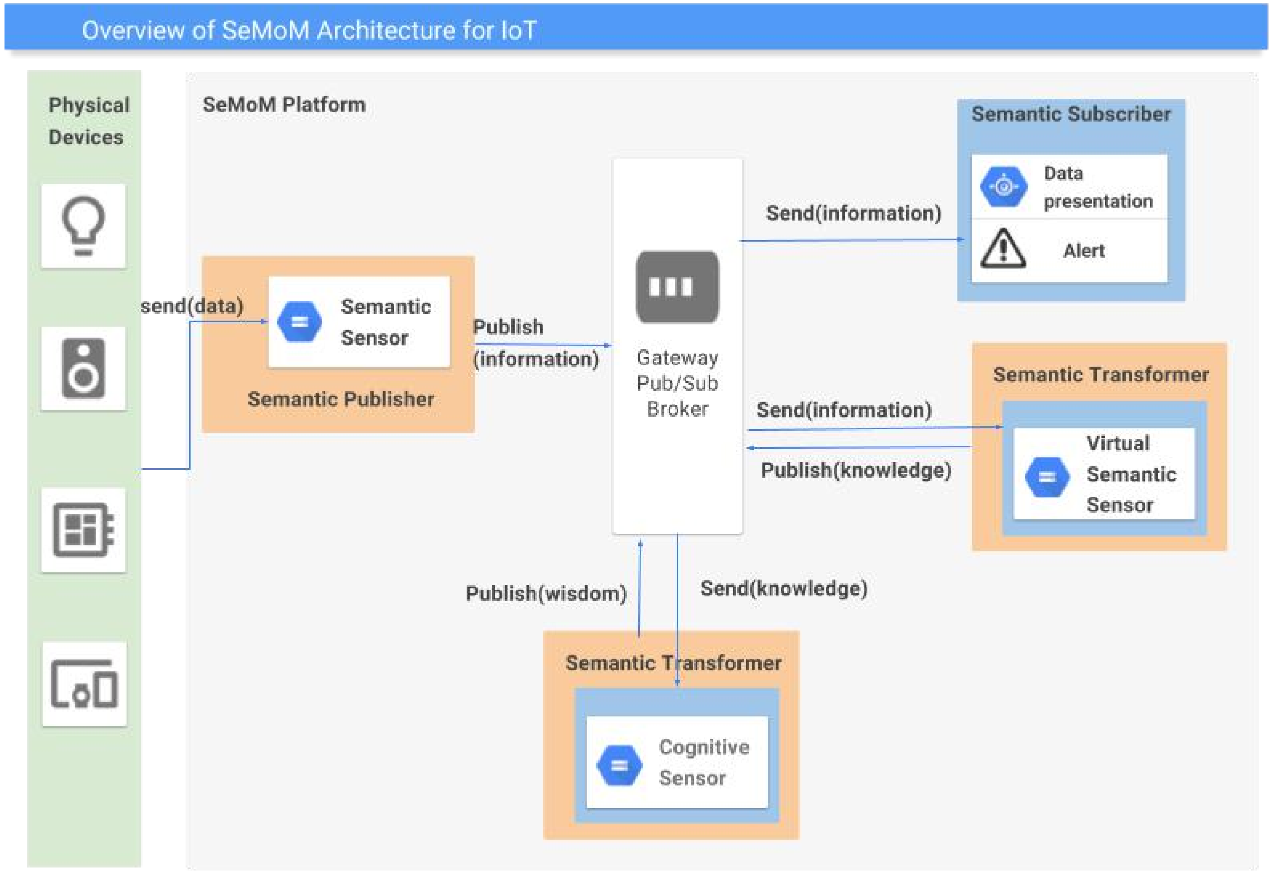


Figure 4.5 Overview of the SeMoM Architecture for IoT Applications

We present in Figure 4.5 the SeMoM architecture with its main components. For each CoSSN component a role from MoM concepts is assigned in such way:

Semantic sensor is a software component that converts raw data into semantic data by CoSSN means. In SeMoM, the semantic sensor requests physical sensors **data**, represents them as OWL classes using CoSSN ontology and publishes the performed **information** to the broker. Hence, the semantic sensor is a semantic data producer and has the **semantic publisher** role in our architecture.

The **virtual semantic sensor** in CoSSN is a component that adds knowledge and domain context to the semantic sensor information. The role of the **virtual semantic sensor** can be manifested in the simple transformer role in MoM. In fact, in SeMoM the virtual semantic sensor subscribes to sensor data or **semantic topic** via the broker, it analyzes the received information and generates new enriched messages as knowledge. Hence, it is in charge to publish knowledge to the broker. Therefore, a virtual semantic sensor plays the double role of semantic publisher and semantic subscriber, in a one term it has the role of **semantic transformer**.

As the **cognitive sensor** is a type of virtual semantic sensor, it has the same role as the **semantic transformer** in SeMoM architecture. The only difference is that it subscribes to many semantic topics via the broker, performs **complex reasoning** (algorithms, machine learning, etc.) and combines different sensor data in order to generate **wisdom**. Its role is more akin to an advanced transformer as described in the previous section. We use SemanticTransformer for all classes of transformers for simplicity.

Analogously to the sequence diagram presented for MoM communication model, Figure. 4.6 presents a sequence diagram presenting the new components of SeMoM. In this diagram, it is clear that there is no direct communication between physicalDevice and the broker. In SeMoM, the main data source is the SemanticSensor that publishes over time to the broker. Moreover, similarly to MoM subscriber, a **semanticSubscriber** subscribes to the broker and receives semantic data. The diagram shows also, the bidirectional communication (SemanticTransformer) role of the component VirtualSemanticSensor.

Messages exchanged between SeMoM components can be seen from two perspectives: topic-based from MoM perspective and property-based from CoSSN perspective. Based

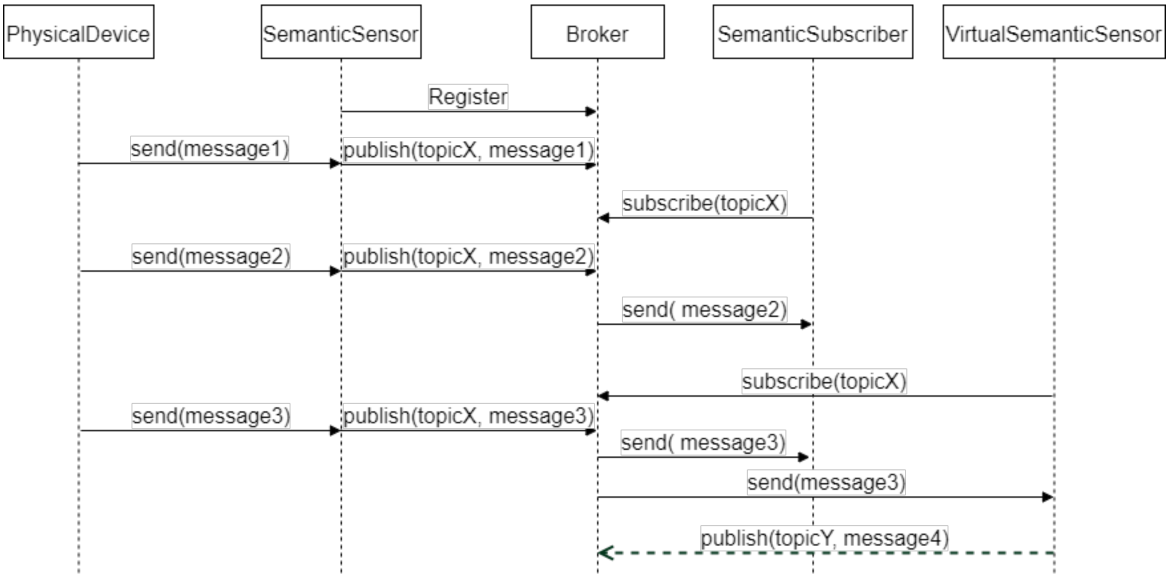
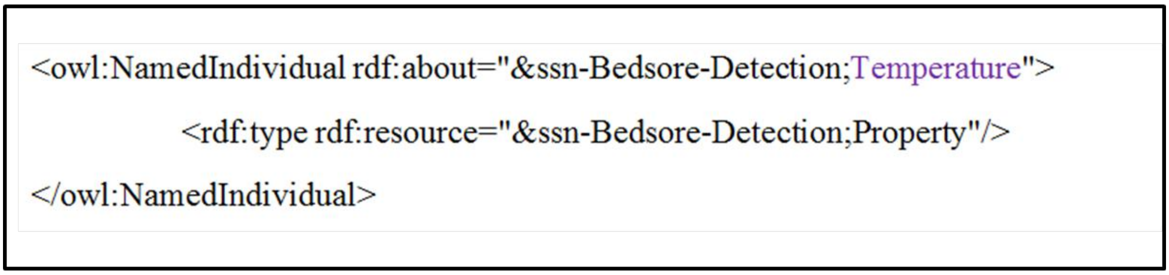


Figure 4.6 Sequence diagram showing SeMoM components and a communication scenario between them

on that, a semantic topic-based mapping is presented in the following showing the consistency of the approach.

4.3.4 Semantic Topic-based Mapping

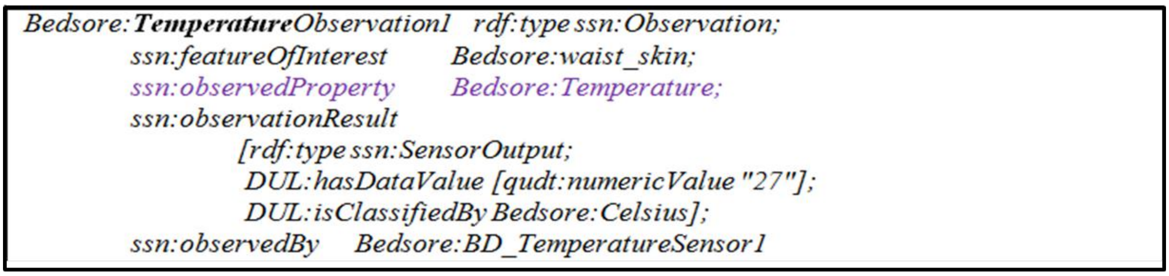
Semantic subscriber represents the receiver component connected to the semantic broker. To receive information, the semantic subscriber has to register for the relevant topic. This topic can be seen as the property entity of the sensor defined in CoSSN ontology. An example of a semantic subscription targeting the temperature topic is illustrated in Figure. 4.7 with the OWL formalism.



```
<owl:NamedIndividual rdf:about="&ssn-Bedsore-Detection;Temperature">
        <rdf:type rdf:resource="&ssn-Bedsore-Detection;Property"/>
</owl:NamedIndividual>
```

Figure 4.7 Example of subscription to topic "temperature" in CoSSN

In the case of an advanced transformer, a component receives different semantic observations for different topics. In order to retrieve specific observations for a specific topic, the component is able to run a SPARQL query specifying the property "ss:observedProperty". This is possible because each observation includes the appropriate specific property as it is shown in Figure. 4.8.



```
Bedsore:TemperatureObservation1   rdf:type ssn:Observation;
          ssn:featureOfInterest       Bedsore:waist_skin;
          ssn:observedProperty        Bedsore:Temperature;
          ssn:observationResult
                    [rdf:type ssn:SensorOutput;
                     DUL:hasDataValue [qudt:numericValue "27"];
                     DUL:isClassifiedBy Bedsore:Celsius];
          ssn:observedBy     Bedsore:BD_TemperatureSensor1
```

Figure 4.8 Example of a publication containing sensor output for the topic "temperature"

A main objective of SeMoM is that its architecture can be extended and specialized for a specific domain. In our work, healthcare has been chosen as an illustration domain.

4.4 Bedsore Risk Detection Illustrative Case Study

To validate the proposed architecture, a prototype for bedsores risk detection has been developed. We have presented in chapter 3 section 3.6 the case study where we have proposed to deploy in the patient's bed or wheelchair the three physical sensors pressure sensor, moisture sensor and a pedometer. The aim of the bedsore application is to gather sensor data, apply the braden scale assessment and then to send a notification to a nurse informing her about patients that are at risk. We will use the same scenario where the patient430 is at a high risk of bedsore occurrence.

As presented in Figure. 4.9, *semantic sensors* receive raw data from physical sensors, use CoSSN to generate *semantic observation* and publish the performed *information* to the broker. For instance, the information sent are:

- "the pressure value is 30 in the bed of Patient430 at time" sent by SemanticPressure sensor for topic "friction".
- "the moisture value is 430 in the bed of Patient430 at time" sent by SemanticMoisture sensor for topic "moisture".
- "the number of steps value that Patient430 did until time is 10" sent by SemanticPedometer for topic "steps".

Information sent by semantic sensors is a domain-independent information that can be used by any consumer/transformer application connected to the broker. For instance two applications presented as *VirtualSemanticPedometer1* and *VirtualSemanticPedometer2* have subscribed to the broker for the topic "steps". These two applications receive information about "steps" and each one applies specific rules and then, generates different knowledge about the meaning of these steps.

- *VirtualSemanticPedometer1* monitors the activity of the person. It analyzes the information sent by the *semanticPedometer* and publishes a new message for the topic "activity". The message contains "the score of the activity factor is 2for the patient430" based on braden scale description.
- Similarly, *VirtualSemanticPedometer2* analyses the semantic information and publishes different knowledge to the broker for the topic "mobility". The message contains "the score of the mobility factor is 2", which means that the Patient430 has a very limited mobility.

The Bedsore Risk Detection application is defined in CoSSN as a cognitive sensor. It connects to the broker for the several topics needed to perform Braden scale moisture, friction, activity and mobility. Hence, the application needs some information from semantic sensors (moisture and friction) and some knowledge (activity and mobility) from the virtual semantic sensors previously defined. Therefore, bedsore application will apply transparently the braden scale score and will output the risk of bedsore. Bedsore application takes then the role of a semantic publisher and send the correlated message to the broker for the topic *bedsore*.

Finally, the nurse via her mobile tablet subscribes to the *bedsore* disease as a semantic subscriber. She receives over time notification about patients such as "Patient430 is at high risk of bedsore occurrence " that is generated by the Bedsore Risk Detection Cognitive Sensor.

It should be noticed that the nurse is able to subscribe to a number of patients in different floors or a specific patient by specifying it in the subscription. It should be also noticed that the SeMoM architecture reflects a full flexible architecture where all applications collaborate transparently.

Furthermore, an application can decide whether to share information to others or not. It can be a simple semantic subscriber that just receives information without sharing it or to be a semantic transformer. In a domain like healthcare, some information should not be shared for privacy issues. In that case the knowledge expert defines the type of communication he desires.

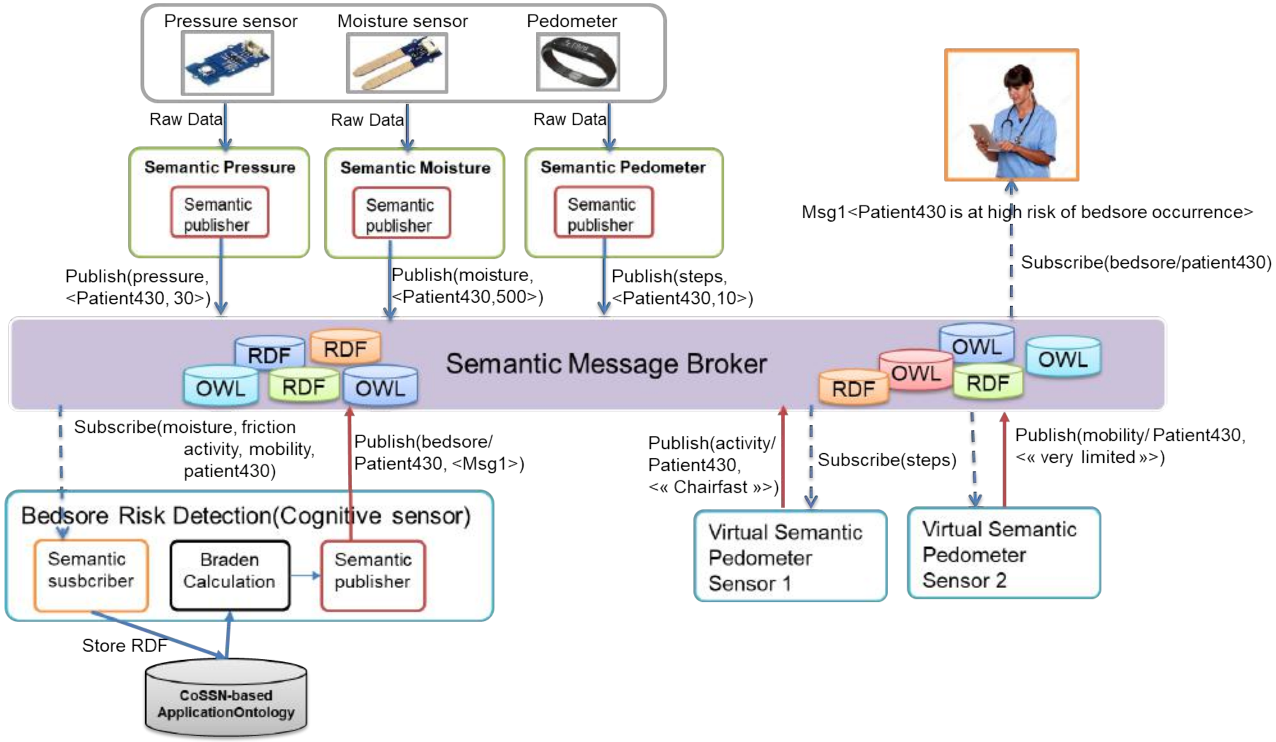

Pressure sensor
Moisture sensor
Pedometer
Raw Data
Raw Data
Raw Data
Semantic Pressure
Semantic publisher
Semantic Moisture
Semantic publisher
Semantic Pedometer
Semantic publisher
Publish(pressure, <Patient430, 30>)
Publish(moisture, <Patient430,500>)
Publish(steps, <Patient430,10>)
Msg1<Patient430 is at high risk of bedsore occurrence>
Subscribe(bedsore/patient430)
RDF
RDF
OWL
RDF
OWL
Semantic Message Broker
OWL
OWL
OWL
RDF
RDF
Subscribe(moisture, friction activity, mobility, patient430)
Publish(bedsore/ Patient430, <Msg1>)
Publish(activity/ Patient430, <« Chairfast »>)
Subscribe(steps)
Publish(mobility/ Patient430, <« very limited »>)
Bedsore Risk Detection(Cognitive sensor)
Semantic susbcriber
Braden Calculation
Semantic publisher
Virtual Semantic Pedometer Sensor 1
Virtual Semantic Pedometer Sensor 2
Store RDF
CoSSN-based ApplicationOntology

Figure 4.9 Bedsore risk detection illustrative example of SeMoM

4.5 Conclusion of the Chapter

In this chapter, we have presented existing middleware approaches that consider semantic description and we showed that none of the proposals can tackle all IoT healthcare requirement. We presented the semantic web of things approach and its benefits towards making the IoT smarter. We presented our design principles that rely on the web of things paradigm. Moreover we presented the SeMoM architecture for distributed IoT applications which implements the semantic web of things paradigms. SeMoM relies on message oriented middleware that is enhanced by semantic description of sensor data and metadata. This combination enables the following features:

- Collecting sensor data from heterogeneous sensors deployed in a smart place;
- Generating a semantic integration, inference and representation of sensor data;
- Distributing the semantic knowledge and wisdom between all system components.

Furthermore, we illustrated the use of SeMoM with by a healthcare application the bedsore application. This use case has been studied and implemented according to SeMoM principles. However, we could not validate clinically this application due to several french laws related to the patients. In order to validate our architecture, we chose to implement the activity detection case study which does not need a direct contact with patients. In the next chapter, we present the technical part and the implementation of SeMoM.

Chapter 5

Human Activity Detection in AAL: A Case Study

"The advance of technology is based on making it fit in so that you don't really even notice it, so it's part of everyday life."

— *Bill Gates*

Contents

5.1 Introduction

In the previous chapter, we presented and highlighted the principles of our SeMOM architecture but an implementation and evaluation of the architecture is still mandatory. Monitoring Activities of Daily living (ADL) is a trending application that researchers try to develop in order to follow elderly at home. This application is characterized by the deployment of several sensors, applying reasoning on sensor data and finally by detecting activities or habits of a person at home. In light of these specifications, our objective of this chapter is to provide a SeMoM-based framework able to detect ADL of elderly. For that, an extension of the CoSSN ontology presented in chapter 3 and an extension of the SeMoM architecture presented in chapter 4 are mandatory to describe the application.

In order to reach our goal from this chapter, we provide in section 5.2 an overview of the existing enabling technologies that serve as a background of our technical choices to implement SeMoM. Then, We present the recognition of activities of daily living. This case study in the AAL domain was conducted to provide a proof-of-concept and to illustrate the benefits and limitations of our proposal. To this aim, we propose a sensors-based framework for human activity recognition based on the CoSSN knowledge description and MoM communication concepts. The SeMoM framework with experimentations performed in real smart-place are presented in section 5.3. Another implementation of SeMoM based on ADLSim, a simulator that generates activities of daily living is presented in section 5.4.

5.2 Overview of Enabling Technologies

Hardware platforms (e.g., microcontrollers, microprocessors, etc.) and software applications represent the computational ability of the IoT. In order to have an overview of the possible technical choices to deploy an IoT system, we reviewed enabling technologies that have been used to develop IoT environments and we summarize them in Table 5.1. This review has served as a background of our deployment in section 5.3.3.

From a hardware perspective, various hardware platforms were developed to run IoT applications such as Raspberry PI, Arduino, UDOO, FriendlyARM, Intel Galileo, Gadgeteer, BeagleBone, Cubieboard, Z1, WiSense, Mulle, and T-Mote Sky. These platforms enable the plugging and manipulation of physical sensors (wearable, ambient, actuators or RFID tags) through communication technologies such as Wifi and Bluetooth for instance. It is likely preferable that a hardware platform provides communication abilities to all communication protocols. Rapberry PI and Arduino are the most used platforms where several physical sensors can be plugged and manipulated.

From a software perspective, several operating systems, software applications and scripting languages have been presented in technical projects for IoT. Several Operating Systems (OS) are utilized to provide IoT functionalities and to develop real-time IoT applications. For instance, Contiki RTOS [222] has been used widely in IoT scenarios. It allows developers to simulate IoT sensors and sensors network. We found also other OS designed for IoT environments like TinyOS [223], LiteOS [224] and Riot OS [225]. Hardware and software applications solutions support several programming languages. C, Java and Python are the preferable scripting languages for IoT applications.

Table 5.1 Overview of IoT implementation technologies

IoT Elements		Technologies
Identification	Naming	EPC, uCode
	Adressing	IPv4 , IPv6
Sensing		Ambient sensors, wearable sensing devices, Actuators, RFID tag
Communication		WiFi, Bluetooth, RFID, NFC, IEEE802.15.4, Z-wave, LTE-A, UWB
Computation	Hardware	Arduino, UDOO, FriendlyARM, Intel Galileo, Raspberry PI, Gadgeteer,BeagleBone, Cubieboard, Z1, WiSense, Mulle, T-Mote Sky, SmartThings
	Software	OS (Contiki, TinyOS, LiteOS, Riot OS, Android); Cloud (Nimbits, etc.)

In order to leverage the full potential of IoT, the interconnected devices need to communicate using lightweight protocols with minimum use of CPU resources. In that context several data exchange protocols have been implemented for IoT environments. They can be classified as message-centric or data-centric based on the study in [226]. Message centric protocols such as MQTT and JMS focus on the delivery of the message regardless of the data payload it contains. Data-centric protocols such as DDS, CoAP and XMPP focus on the delivery of data and assume that data is understood by the receiver.

- MQTT [227] : The Message Queuing Telemetry Transport (MQTT) is a lightweight publish-subscribe messaging protocol used in IoT application that has been standardized by OASIS since 2014. It is based on a message broker that uses TCP connection and serves as a mediator between participants. It also supports hierarchical topics like "subject/sub-subject".
- HTTP REST [228]: It represents a simple way to exchange data between clients and servers over HTTP. It is used within mobile and social network applications and it eliminates ambiguity by using HTTP get, post, put, and delete methods. It allows to implement web service applications and RESTful implementations make use of standards, such as HTTP, URI, JSON, and XML.
- DDS [229]: It is a data-centric middleware based on publish-subscribe protocol. It has been developed by the Object Management Group (OMG) for scalable and real-time M2M communication. It is broker-less protocol and supports UDP, TCP/IP and IP Multicast transport protocols to send and receive data.
- CoAP [230]: It is a client server model transfer protocol that has been designed for constrained devices. It uses UDP connection and supports broadcast and multicast addressing. CoAP supports content discovery that allows devices to find each others and to exchange data. Moreover, it supports asynchronous communications and RESTful protocol.
- XMPP [231]: XMPP is an IETF instant messaging standard that is used for multi-party chatting, voice and video calling and telepresence. It allows users to communicate with each other by sending instant messages on the Internet no matter which operating system they are using. It connects a client to a server using a stream of XML data which represents a code divided into three components: message, presence, and iq (info/query).
- AMQP [232]: It is an open standard protocol for IoT focusing on message oriented communications. It supports reliable communication via message delivery guarantee primitives including at-most-once where each message is delivered once or never, at-least-once where the message is certain to be delivered but it may

arrives multiple times, and exactly once delivery where the message will certainly arrive only once.

5.3 Sensors-based ADL Detection System

Aging population, reducing the cost of health care, and the importance that individuals place on remaining independent in their own homes are the motivation of most activity recognition researches in smart environments. An activity detection system automates the recognition of Activities of Daily Living (ADLs) such as eating, grooming, cooking, drinking and taking medicine. It also allows to track these activities toward monitoring the functional health of a smart home resident which has also been recognized by family and caregivers of Alzheimer's patients [41].

In the context of activity recognition [40] [41], motion, force, vibration, water, buttons and other sensors provide individual state information. Information gathered from each sensor is then merged and correlated in order to recognize the activity of the person. For example, meal preparation, sleeping, watching tv, taking a snap or eating are such activities. The detection of ADL allows to provide valuable information from a medical standpoint about the inhabitant's behavior [37].

5.3.1 Overview of ADL Recognition Systems

In the last few years, several solutions have been provided for human activity detection. Video-based or vision-based approach is one of the most used technique to do so relying on visual surveillance, video retrieval or human-computer interaction techniques to detect activities. However, elders consider cameras as intrusive and privacy invasive thus limiting their acceptance. Deploying cameras and sound sensors often instills a disruption and uncomfortable feeling [33] [30] [31] [32]. Other approaches provide monitoring features by using different kinds of sensors (wearable, smartphone-based or ambient) and several techniques to parse a stream of sensor data so that human activities can be inferred. Wearable or carried sensors have been widely used over the year to monitor physical activities such as in [37] and to detect the according activities such as in [34] and [35].

Moreover, middleware solutions for ADL recognition have been also proposed. For example, [233] introduces a middleware that allows for the recognition of activities and indoor mobility to support context-aware services. A context-driven approach was developed in [234] for profiling the user's activity at multiple levels of granularity. The system combines motion reconstruction, location detection and activity identification to describe the user's context. In [235], the authors present a platform to gather users' psychological, physiological and activity information for mental health purposes.

In order to infer and detect activities, IoT applications solutions use artificial intelligence algorithms and techniques. A large number of existing solutions rely on data mining and machine learning techniques [67] which have been applied to learning associations between factors and correlations of relevant contexts. These methods that are referenced for ADL detection span a broad range of techniques from simple algorithms to more advanced methods like hidden Markov models (HMMs) with several of its extensions, Bayesian networks, decision trees, neural networks and so on. The most distinguishable features of machine learning techniques is their ability to correlate and combine sensor data and situation and to extract categorical activities such as running or walking for instance. However, these techniques usually require a long period of learning before figuring out the rules that detect the activities.

Another important aspect of existing proposals is the use of semantics and knowledge-driven models for ADL descriptions. Ontologies in this strand have been widely discussed and many ontologies-based solutions have been proposed [70], [71], [72]. With respect to other solutions, knowledge-based methods are semantically clear in modeling and representation and highly effective in inference and reasoning. They create a complete accurate model of ADL [73] that enhances the semantic interoperability of an IoT system whitout a prior learning phase. However, the aforementioned solutions describe activities but there is a lack in representing and describing sensors in the same ontology.

5.3.2 SeMoM Framework based on Physical Sensors

ADL detection systems are witnessing a high revolution in the research domain. Many attempts have been proposed and it still need more improvements. In this thesis, we do not deal with the problematic of ADL detection itself. However, the active research in this field and the fact that we can deploy it, motivated us to acquire this case study in order to implement and evaluate our SeMoM architecture. Hence, CoSSN ontology will be extended to represent the ADL application with semantic and virtual sensors that will be defined as well. Moreover, SeMoM framework presented in Chapter 4 will be extended to create the ADL application.

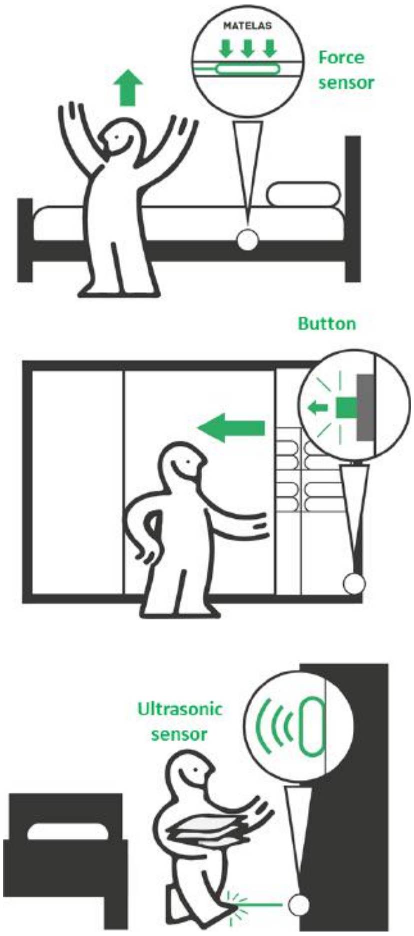


Figure 5.1 Illustration of the "Dressing Up" activity.

5.3.2.1 Description of the Framework

In our framework, activity detection is a type of cognitive sensing with the aim to monitor predefined ADLs based on data from physical sensors in near real-time. Semantic descriptions of such activities and of their relationship with semantic sensor data through the actions and/or other activities, allow to implement activity detection in a modular way. Any specific activity or action detection that is required by other activity detection applications could be deployed as an instance of cognitive sensor in the SeMOM architecture thus improving system maintenance and evolution by code optimization and reuse.

More specifically, we focus on the "Dressing Up" activity in order to present our framework but, in order to describe other activities the same process can be followed.

As it is shown in Figure. 5.1, the "Dressing Up" activity can be detected by aggregating three actions: 1) the person gets up from his bed, 2) the person opens the wardrobe and 3) detecting and verifying the presence of the person in his bedroom. Based on that, five main components are defined to detect the activity as presented in Figure. 5.2. These components are:

1. Physical sensors: we assume that the "Dressing Up" activity can be inferred by detecting the three actions as illustrated in Figure. 5.1. In order to detect these actions three physical sensors are deployed: force sensor to detect the action number 1, button sensor to detect the action number 2 and ultrasonic sensor to detect the action number 3.

2. Semantic sensors: as described in chapter 3, each physical sensor can be represented in the CoSSN ontology. In the same way, each mentioned physical sensors are described via CoSSN. The semantic sensors publish semantic observations to the broker for a specific property. For example, semanticForce publishes the value 3 for the topic "force".

3. Actions sensors: are developed as virtual semantic sensors. Each one subscribes to a specific topic in order to infer the actions of the person. In the case presented in Figure. 5.2, the *VirtualBedSensor* subscribes to "force" topic and infers that the person is not in bed then sends the value "False". The *VirualPresenceSensor* subscribes to "presence" topic and generates the observations indicating that the person is in the room. And finally the *VirtualWardrobeSensor* subscribes to button topic and infers that the "button" is pressed.

4. Dressing Up Sensor: is the cognitive sensor that subscribes to the three topics: "wardrobeOpened", "personInBedroom" and "personInBed". It receives the semantic observation from actions sensors and store it in CoSSN-based Application Ontology. The aim of this cognitive sensor is to treat and to interpret sensor data provided by producers in order to infer new semantic information and content-rich knowledge. The inferred data can be computed by applying decision or rules such as artificial intelligence techniques. For instance, a SPARQL query module is implemented in the cognitive sensor component that aims to detect the human activity recognition such as ”dressing up activity”. After detecting the activity, the Dressing Up sensor component sends back to the broker the new semantically enriched information depicting the Dressing up activity in compliance with the semantic publisher role previously defined.

5. Nurse Application: finally, we define, an application dedicated to the medical staff. The application is defined as a semantic subscriber that subscribes to the activity of the person. It receives near real-time notifications when a critical state of a monitored person occurs and requires medical intervention.

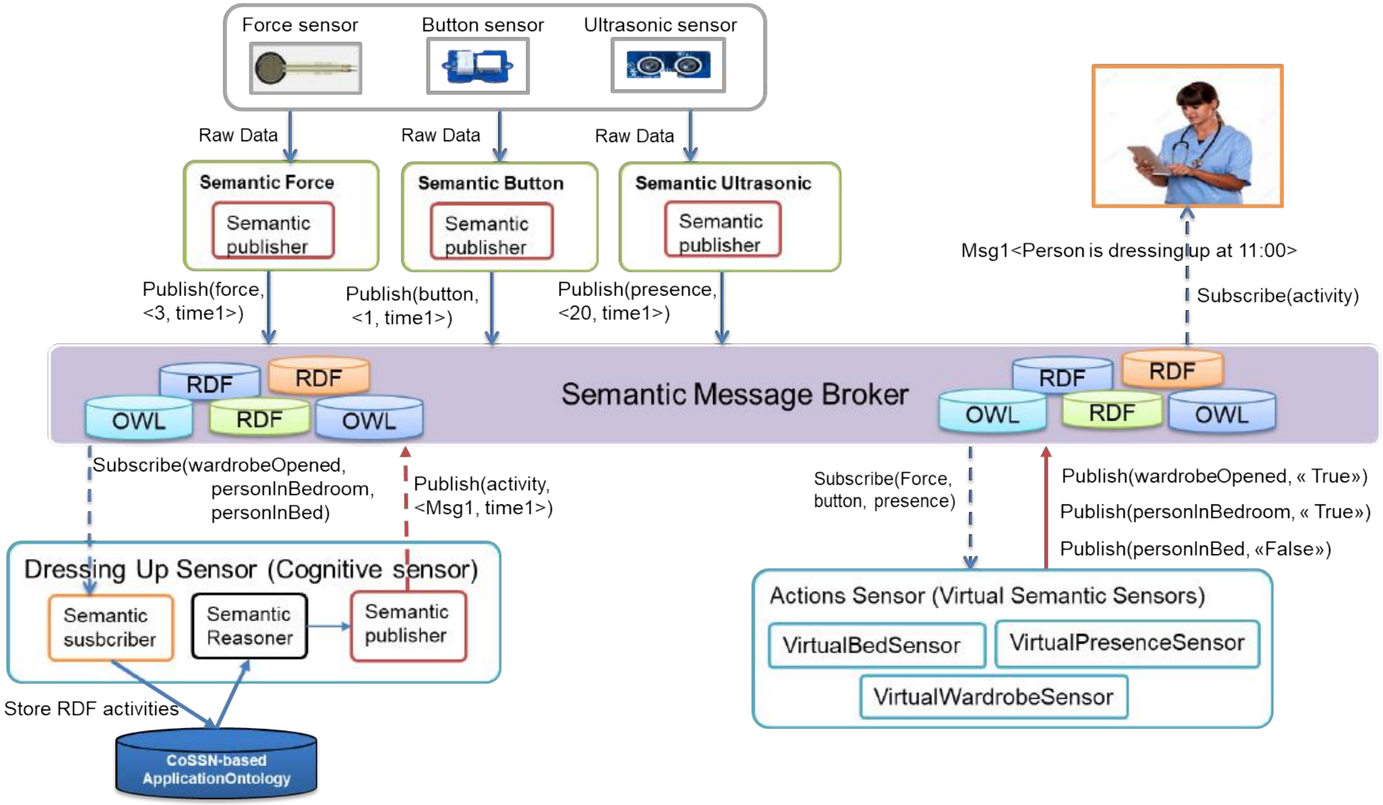
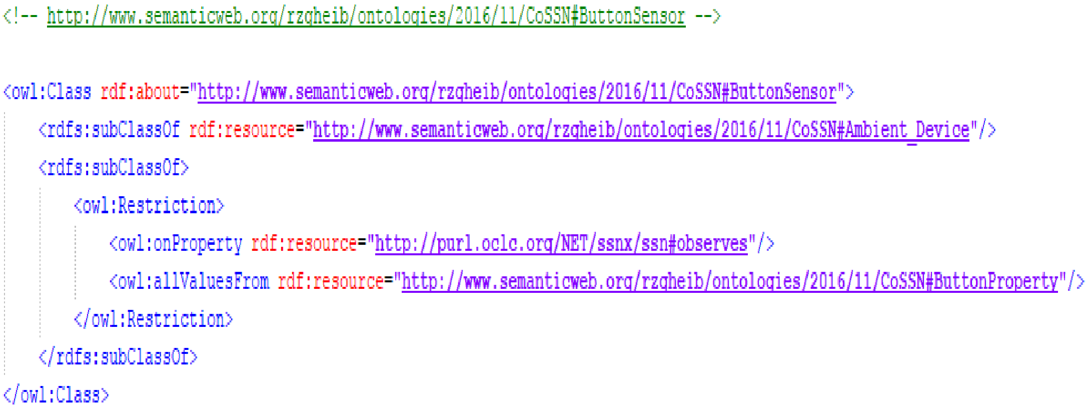

Force sensor
Button sensor
Ultrasonic sensor
Raw Data
Raw Data
Raw Data
Semantic Force
Semantic publisher
Semantic Button
Semantic publisher
Semantic Ultrasonic
Semantic publisher
Publish(force, <3, time1>)
Publish(button, <1, time1>)
Publish(presence, <20, time1>)
Msg1<Person is dressing up at 11:00>
Subscribe(activity)
RDF
RDF
OWL
RDF
OWL
Semantic Message Broker
RDF
RDF
OWL
RDF
OWL
Subscribe(wardrobeOpened, personInBedroom, personInBed)
Publish(activity, <Msg1, time1>)
Subscribe(Force, button, presence)
Publish(wardrobeOpened, « True»)
Publish(personInBedroom, « True»)
Publish(personInBed, «False»)
Dressing Up Sensor (Cognitive sensor)
Semantic susbcriber
Semantic Reasoner
Semantic publisher
Actions Sensor (Virtual Semantic Sensors)
VirtualBedSensor
VirtualPresenceSensor
VirtualWardrobeSensor
Store RDF activities
CoSSN-based ApplicationOntology

Figure 5.2 High-Level ADL detection system architecture

5.3.2.2 Application of the CoSSN Ontology for ADL description based on Physical Sensors

The before mentioned components have been described in the CoSSN ontology. This is achieved by first extending the CoSSN ontology described in Section 3.4 with application specific concepts as follows.

- **Application specific sensors**. Based on the human activities to be detected, the physical sensors are identified and each one is described as a specialization of *Ambient_Device* entity of the CoSSN ontology to represent its semantic wrapper. For example, **Force**, **Button** and **Ultrasonic** sensors of Figure 5.2 are described by **ForceSensor**, **ButtonSensor** and **UltrasonicSensor** entities in the ontology. The property each sensor is observing is defined as a subclass of the *Property* entity of the SSN ontology. Following the example of Figure 5.2, **ForceProperty**, **ButtonProperty** and **UltrasonicRangerProperty** are added in the ontology, and the *observes* object property is specialized accordingly, e.g., **ForceSensor** *observes* **ForceProperty**. The property types will be used to semantically annotate the topics of the publish/subscribe communication protocol to the purpose of semantic interoperability of sensors. An example of button sensor description is presented in Figure. 5.3, the other sensors description follow the same concept.



```
<!-- http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#ButtonSensor -->

<owl:Class rdf:about="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#ButtonSensor">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#Ambient_Device"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://purl.oclc.org/NET/ssnx/ssn#observes"/>
            <owl:allValuesFrom rdf:resource="http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#ButtonProperty"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 5.3 An excerpt of the RDF triple describing the Button sensor

- **Application specific virtual and cognitive sensors**. Similarly to physical sensors, virtual sensors used for the specific human actions detection and the cognitive sensor used for activity detection can be described as specializations of *VirtualSensor* and *CognitiveSensor* entities of CoSSN. A taxonomy of two sensors is defined through subsequent specializations, namely **ActionDetectionSensor** for detecting human action types cognitive stimuli and **ActivityDetectionSensor** for activity types. The specific types of action/activity detection sensors defined in the ontology actually map to different functionalities of common action/activity detection software components.

As we have presented in the architecture of Figure. 5.2 two types of sensors are deployed. **Actions Sensor** is a software component devoted to detecting human actions from sensors annotated data and **DressingUpSensor** is devoted to detecting the human activity represented by *the person is dressing up* cognitive stimulus in the ontology. For this example, such stimulus is detected on the basis of the following other cognitive stimuli: *a person is up*, *a person is in the room*, and *a person is opening the wardrobe*, which are detected by **VirtualBedSensor**, **VirtualPresenceSensor** and **VirtualWardrobeSensor** respectively. Similarly to physical sensors, each specific ActionDetectionSensor observes a specific action, e.g., **VirtualBedSensor** *observes* **PersonInBedProperty**, a specialization of *ssn:Property* as it is presented in Figure. 5.4. Instead, each ActivityDetectionSensor observes a specific activity, i.e., **PersonDressingUpDetectionSensor** *observes* **PersonDressingUpProperty**, a specialization of **ActivityProperty**, which, in turn, is *a proxy for the person is dressing up* cognitive stimulus.

- **Application specific actions and activities**. The specific activities and actions to be detected are specified by specializing respectively Activity and Action entities of CoSSN. Furthermore, each action type can be derived from one or more stimuli as we have presented in section 3.4.2.1 in Figure. 3.15, e.g., **PersonInBed** action sub-entity can be *derived from* the *ultrasonic* stimulus. Similarly, each activity type can be detected by means of one or more action (or activity) types cognitive stimuli, as already shown for *the person is dressing up* activity. A cognitive sensor observes a property by applying a **CognitiveSensing** method. For action and activity detection, such a method is defined by a logical formula. For example, **PersonUpProperty** can be observed through the rule ($force < 5$), **PersonInRoomProperty** through ($ultrasonic < 20$), and **WardrobeOpeningProperty** through ($button = 1$). Analogously, **PersonDressingUpProperty** is observed through the formula ($PersonUp = true \wedge PersonInRoom = true$

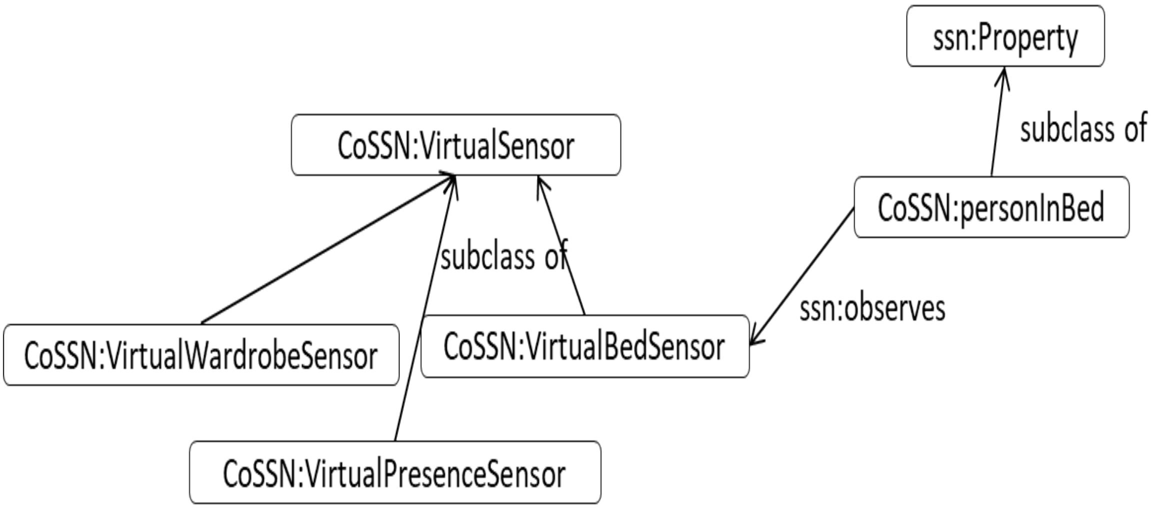


Figure 5.4 An excerpt of the CoSSN representation of virtual sensors

$\wedge$ *WardrobeOpened* = *true*). Such properties can be defined through SPARQL statements.

The cognitive sensing method of each cognitive sensor, associated with an action or activity, is implemented by a SPARQL query that is retrieved, (possibly dynamically) instantiated and run by the corresponding software application as it is shown in Figure. 5.5.

5.3.3 Experimental Environment

In order to deploy our framework several technologies and libraries have been used. They can be classified into three main technologies: hardware technologies, semantic libraries and finally middleware technologies.

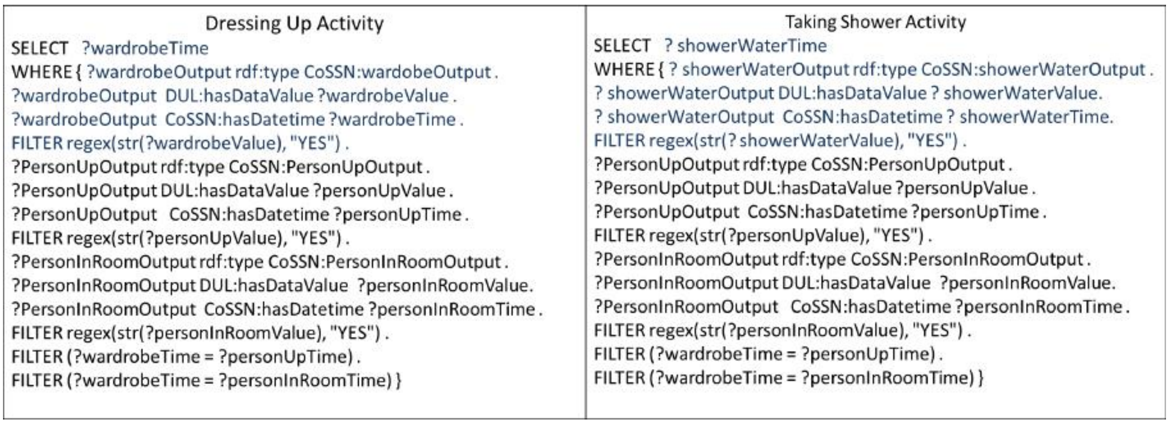
For physical sensors, we chose to deploy Grove sensors [1] since over 100 Grove sensors are available in the market such as "Grove-Temperature&Humidity Sensor" and "Grove-Button". They are easy to connect to a raspberry pi allowing quick deployment for prototyping purposes. Moreover, there exists a Java working library JGrove to manipulate these sensors through Java. Other sensors like a FSR force sensor were implemented using the basic RXTX library. This sensor has been connected to an arduino board. Hence, at this level we used raspberry pi, arduino and different kinds of

[1] https://www.seeedstudio.com/category/Sensor-for-Grove-c-24.html

sensors. This heterogeneity of technologies has been chosen on purpose to emphasize the versatility of our architecture.

SPARQL is an RDF Query Language and a W3C Recommendation since 2008. It provides a set of analytic query operations to create and apply rules on data that follows the RDF specification at near real-time. The choice of SPARQL was made based on the fact that we are using ontologies and exchanging RDF triplets over the middleware system components. Regarding the description of data, the OWL ontology is developed and tested in Protégé 5.0.0 [2] which allows the creation and inferrence of equivalent classes as well as the execution of SPARQL rules.

For Message oriented communication, we chose MQTT protocol since it is a lightweight publish-subscribe messaging protocol used for IoT applications which has been standardized by OASIS in 2014. To implement the framework, we used Java as coding language. This choice is made considering the availability of a large choice of libraries: RDF4J that allows us to model, to manipulate ontologies and to implement SPARQL rules as well; and Paho MQTT Client library that allows to communicate via a MQTT broker.



Dressing Up Activity

```
SELECT   ?wardrobeTime
WHERE { ?wardrobeOutput rdf:type CoSSN:wardobeOutput .
?wardrobeOutput  DUL:hasDataValue ?wardrobeValue .
?wardrobeOutput  CoSSN:hasDatetime ?wardrobeTime .
FILTER regex(str(?wardrobeValue), "YES") .
?PersonUpOutput rdf:type CoSSN:PersonUpOutput .
?PersonUpOutput DUL:hasDataValue ?personUpValue .
?PersonUpOutput   CoSSN:hasDatetime ?personUpTime .
FILTER regex(str(?personUpValue), "YES") .
?PersonInRoomOutput rdf:type CoSSN:PersonInRoomOutput .
?PersonInRoomOutput DUL:hasDataValue  ?personInRoomValue.
?PersonInRoomOutput  CoSSN:hasDatetime ?personInRoomTime .
FILTER regex(str(?personInRoomValue), "YES") .
FILTER (?wardrobeTime = ?personUpTime) .
FILTER (?wardrobeTime = ?personInRoomTime) }
```

Taking Shower Activity

```
SELECT   ? showerWaterTime
WHERE { ? showerWaterOutput rdf:type CoSSN:showerWaterOutput .
? showerWaterOutput DUL:hasDataValue ? showerWaterValue.
? showerWaterOutput  CoSSN:hasDatetime ? showerWaterTime.
FILTER regex(str(? showerWaterValue), "YES") .
?PersonUpOutput rdf:type CoSSN:PersonUpOutput .
?PersonUpOutput DUL:hasDataValue ?personUpValue .
?PersonUpOutput  CoSSN:hasDatetime ?personUpTime .
FILTER regex(str(?personUpValue), "YES") .
?PersonInRoomOutput rdf:type CoSSN:PersonInRoomOutput .
?PersonInRoomOutput DUL:hasDataValue  ?personInRoomValue.
?PersonInRoomOutput   CoSSN:hasDatetime ?personInRoomTime .
FILTER regex(str(?personInRoomValue), "YES") .
FILTER (?wardrobeTime = ?personUpTime) .
FILTER (?wardrobeTime = ?personInRoomTime) }
```

Figure 5.5 Excerpt of SPARQL query rules used for dressing up and taking shower activities

The framework was deployed in accordance with the deployment diagram presented in Figure.5.6. As described in this diagram, the framework contains two basic components connected to a MQTT broker: (i) *DressingUpCognitiveSensorComponent* (DC)

[2] http://protege.stanford.edu/. Last access on 1st September, 2017.

which implements a SPARQL query module and (ii) *ActionsCognitiveSensorComponent* (AC). The application is launched once the semantic sensor discovers physical devices. For instance, GrovePi and RXTX libraries have been used for this aim through the third implemented component, the *PhysicalSensorsComponent*. Each one of the physical sensors is wrapped with a semantic description based on CoSSN concepts thanks to RDF4J library. Upon connection with the broker, the AC which implements the MQTT_publisher component, "getSemanticObservations", creates the action message and publishes over time semantic observations. For example, publish(wardrobeOpened, "YES") is received basically from Button. As a cognitive sensor component, DC component subscribes to topics "wardrobeOpened", "personInRoom" and "personUp" via the MQTT_Subscriber component which will update the *CoSSN-based Application Ontology* and add the received semantic messages. The ActivityDetectionSparqlQuery module implementing SPARQL query rules, requests the *CoSSN-based Application Ontology* for activity as it is shown in Figure. 5.5. Finally, DC component will use MQTT_publisher to publish the new inferred activity to the broker.

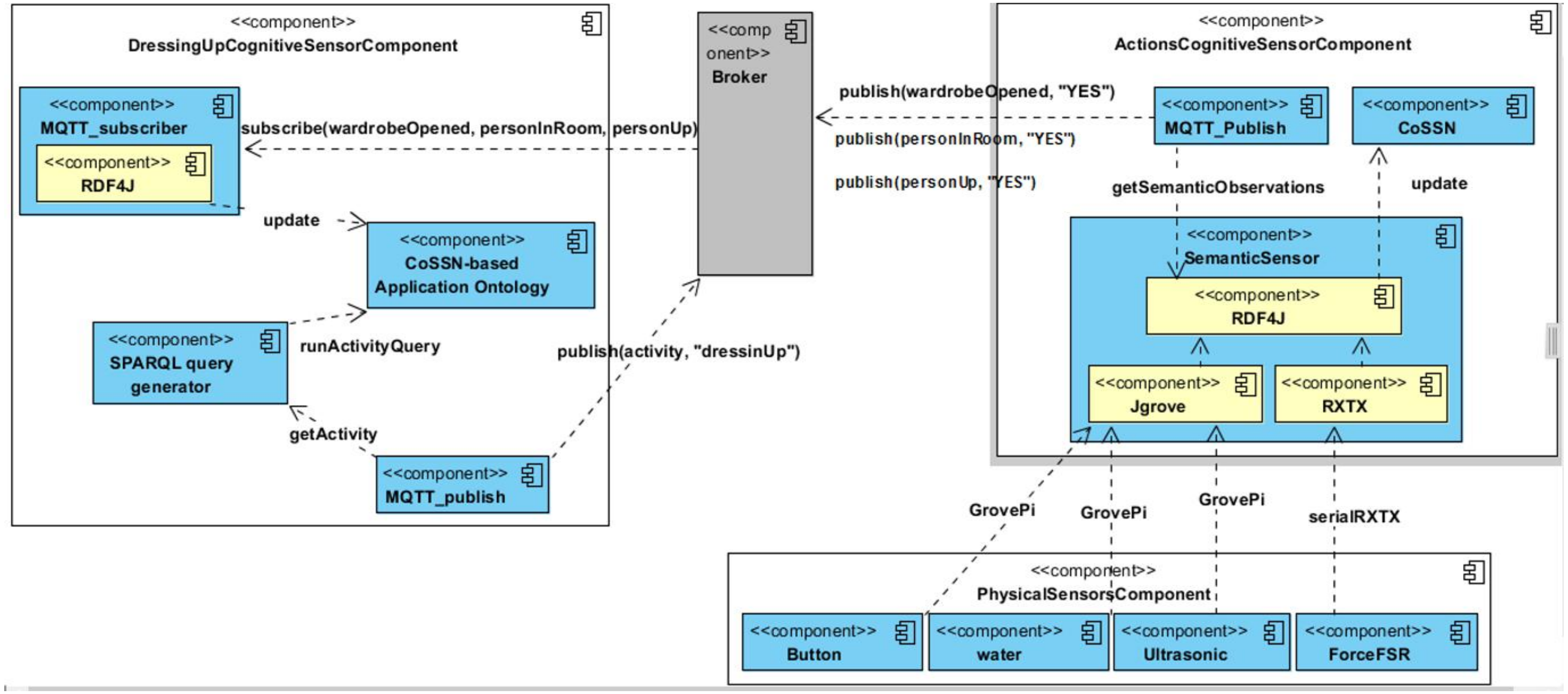


Figure 5.6 Activity detection deployment diagram: An example of SeMoM deployment

Based on the deployment diagram presented in Figure. 5.6, several experiments were carried out with real actors to validate our system. Three main goals have been defined:

1. To implement this framework in a real smart-place and to analyze data from real situations.
2. To gather data from different kinds of ambient sensors with heterogeneous technologies. This heterogeneity of the used technologies allows us to validate SeMoM in providing interoperability through an heterogeneous IoT system.
3. To test the effectiveness and performance of the SPARQL query module implemented for activity detection.

According to the defined goals, we deployed this framework in our experimental platform called the Connected Health Lab (CHL) [3]. It contains an accurate simulation of a patient room, hygiene room and living room. With respect to the "Dressing Up" activity described in the previous section, different sensors have been deployed in this place as follows:

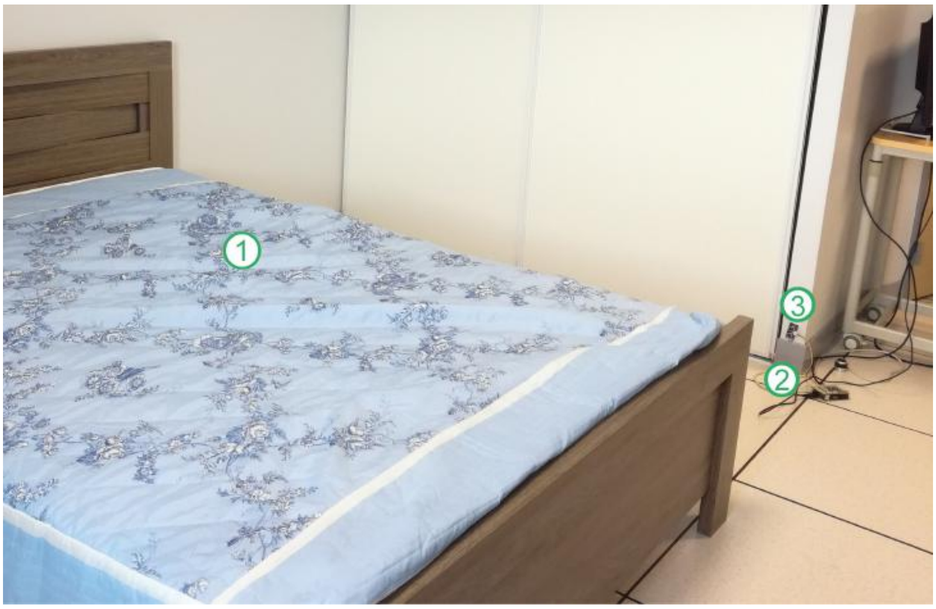


Figure 5.7 Overview of the CHL equipped with physical sensors

Force sensor: A Force Sensitive Resistor (FSR) has been deployed in the bed of the person. This FSR will vary its resistance depending on how much pressure is being

[3] http://chl.univ-jfc.fr/

applied to the sensing area. It is used to detect when the person gets up from bed, and then to infer the action "personInBed". It sends numeric values, when a value is below 10 means that the person has got up.

Ultrasonic sensor: A Grove ultrasonic sensor has been placed between the bed and the wardrobe as presented in Figure. 5.8. The distance between the sensor and the first obstacle will be updated when the person passes by. It is used to detect the presence of the person around the wardrobe. Hence, it can be used to infer the action "personInBedroom".

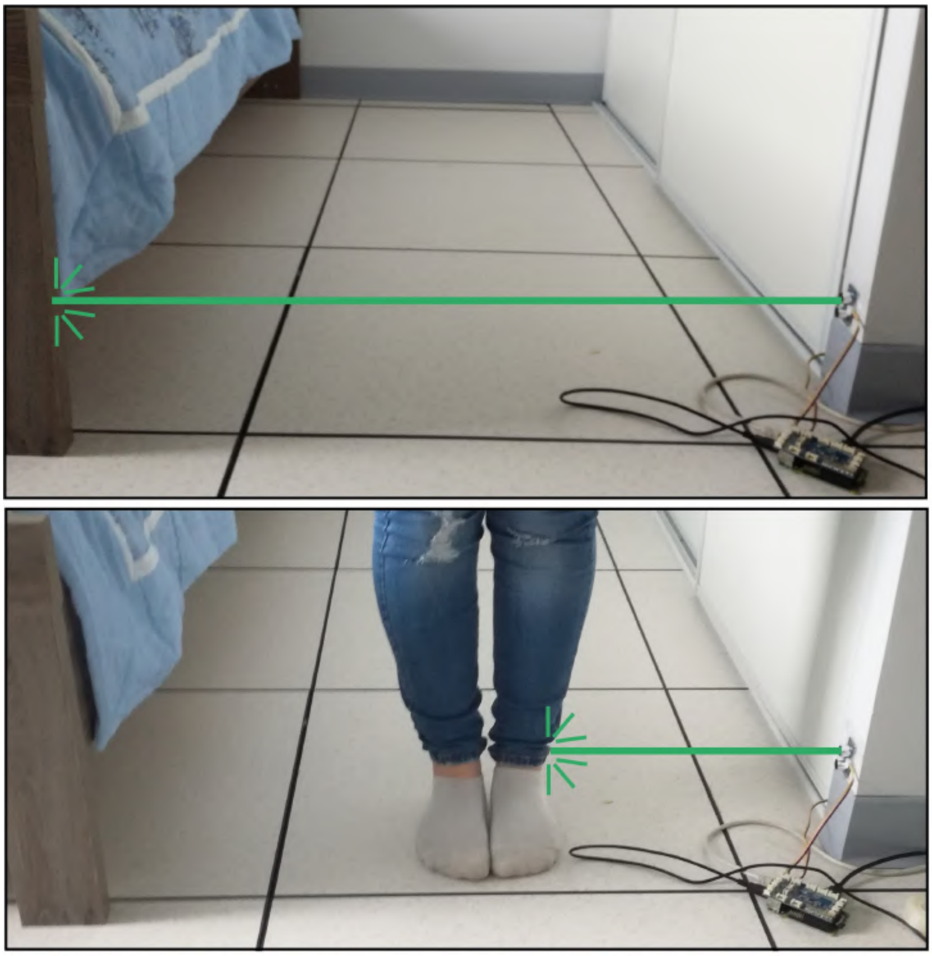


Figure 5.8 Deployment of the ultrasonic sensor in the CHL

Button sensor has been deployed in the wardrobe as depicted in Figure. 5.9. This button on the door of the wardrobe goes from 1 (button pressed) to 0 (button released) when the person opens the door. Hence, it can be used to infer the action "wardrobeOpened".

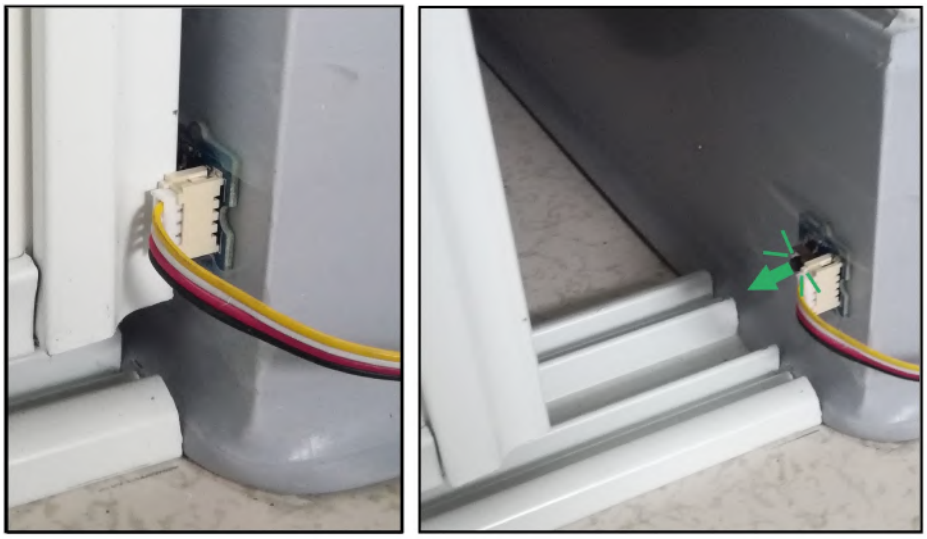


Figure 5.9 Deployment of the button sensor in the CHL

In order to assess the accuracy of the implemented IoT-based application used for cognitive sensing of ADL, the experiments were performed as follow: two actors repeated the three aforementioned activities twice per day for 5 days. As depicted in Table 5.2 the accuracy of detection per actor is 9.33 over 10 times for Actor 1 and 9 over 10 for Actor2. By deploying and running this framework, we have validated and provided a proof of concept for our architecture and we have achieved the defined goals 1 and 2.

As one of the goals of the framework implementation is to assess the performance of SPARQL queries, one of the key validation is to use same sensors to detect different activities. For that, another cognitive sensor component was deployed with the aim to detect "taking shower" activity. Taking shower activity is detected based on 3 actions: Person is not in the bed, person is not in the room, the water shower sensor is ON. In our framework, the detection of this activity relies on the same Force and Ultrasonic sensors used for DressingUp activity but alternatively on a water sensor in the shower. An example of the SPARQL Query used to infer the activity is shown in Figure 5.5.

Results show an accuracy of 85% for "dressing up" and 90% for "taking shower". The difference is due to delay in ultrasonic sensor which has sent data 2s later 2 times with Actor1 and 3 times with Actor2. It is important to notice that our objective is not to assess the accuracy and precision of physical sensors. What matters to us is that these results show that our ontology complies to the objectives we introduced in Chapter

Table 5.2 Activity Detection Experimentations

Room	Actor1	Actor2	Detection
Dressing Up	9/10	8/10	85%
Taking shower	9/10	9/10	90%
Watching TV	10/10	10/10	100%
Average per actor	9.33/10	9/10	91.66%

1 and that SPARQL queries reveal 100% detection when sensor data are received correctly. Therefore, we validate that the framework is able to handle different sensors from different technologies. It is also able to describe physical and virtual sensors for a specific application and to generate domain-specific data based on SPARQL rules. Moreover, results in Table 5.2 suggest that the framework is consistent with the defined concepts of CoSSN and the above mentioned IoT-based application has a good detection accuracy.

5.4 ADLSim-based ADL Detection System

When deploying physical sensors, several types of deviations such as the noise and sensitivity can occur which limit sensor accuracy and generate delayed messages. In light of this limitation and since our capacity of deploying a large number of sensors was not possible at the time of writing, we decided to implement SeMoM framework based on ADLSim simulator. Moreover, this deployment emphasizes that our architecture allows an easy replacement of physical sensors by simulated sensors without affecting the rest of the deployment. It also facilitates the validation of detection algorithms from predefined benchmarks.

5.4.1 Description of the Framework

ADLSim is a simulator that generates a sequence of activities of daily living in the apartment of a monitored person. It has been developed by Stein K. et al from the university of Oslo DMMS department [4]. The current simulation uses the same seven activities of daily living that is used in the widely used Ubicomp dataset by Kasteren et al. obtainable here [5], which are Breakfast, Dinner, GetDrink, GoToBed, LeaveHouse, TakeShower and UseToilet as it is presented in Figure. 5.10 in addition to the activity Idle which occurs whenever nothing else happens. The parametrization of each activity,

[4] http://www.mn.uio.no/ifi/?vrtx=unit-view&areacode=150570

[5] https://sites.google.com/site/tim0306/datasets

i.e., its instigation time, and its duration, is extracted from the Ubicomp trace to reflect the real behavior in that trace. For instance, the duration of the GoToBed activity is specified with a normal distribution with an average of 557.18 minutes and an SD of 74.21 minutes. ADLSim currently simulates 100 days (controlled by numDays), with a time compression factor of 3600 * 24 (controlled by timeCompressionFactor), i.e., one simulated day passes for each real second. The output currently includes a time-stamp with simulated time, an activity ID and an activity name.

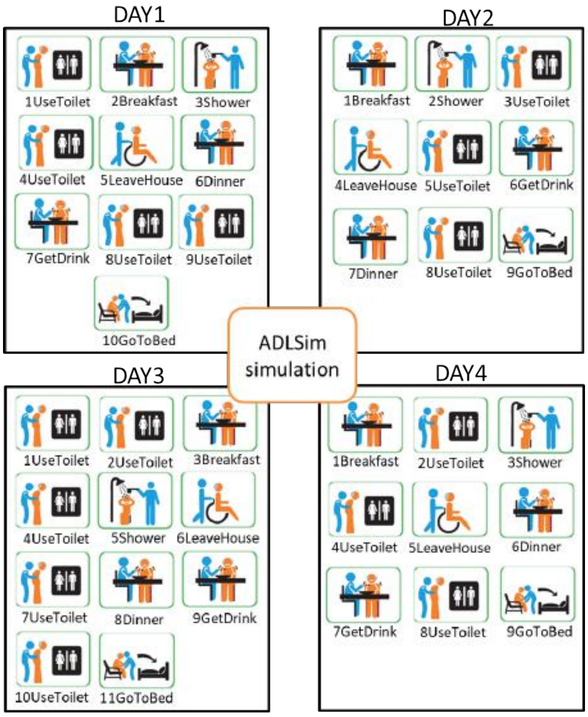


Figure 5.10 Simulation of Four days activities generated with ADLSim simulator

A collaboration with the UiO university has led to use ADLSim for experimental purposes in our study. Based on SeMoM principles, three main components have been defined and described with CoSSN ontology:

1. ADLSim: is defined as a semantic sensor and a semantic publisher that generates semantic observations about the activities of a monitored person. *ADLSim* has been defined as a semantic sensor in CoSSN ontology which observes the *ActivityProperty* as described in Figure.5.11. The feature of interest related to the property is defined as "CoSSN:Thomas" to identify the monitored person. It should be noticed that an instance of ADLSim is able to monitor one person

at home therefore, in order to simulate the monitoring of several patients in different home different instances of ADLSim must be deployed. The precision of the feature of interest enables to identify each monitored person and the ADLSim instance.

2. A statistical application: is defined and developed as a cognitive sensor that subscribes to the "activity" Property defined in CoSSN. The purpose of this application is to infer the daily activities report of a monitored person by applying SPARQL queries. In the case of monitoring several patients in different homes, the application is able to generate the habits and activity diagnostic for each person by specifying the "featureOfInterest" in the SPARQL query.

3. Finally, a simple semantic subscriber is defined to subscribe to the habits of one or more persons and to visualize the inferred and processed information.

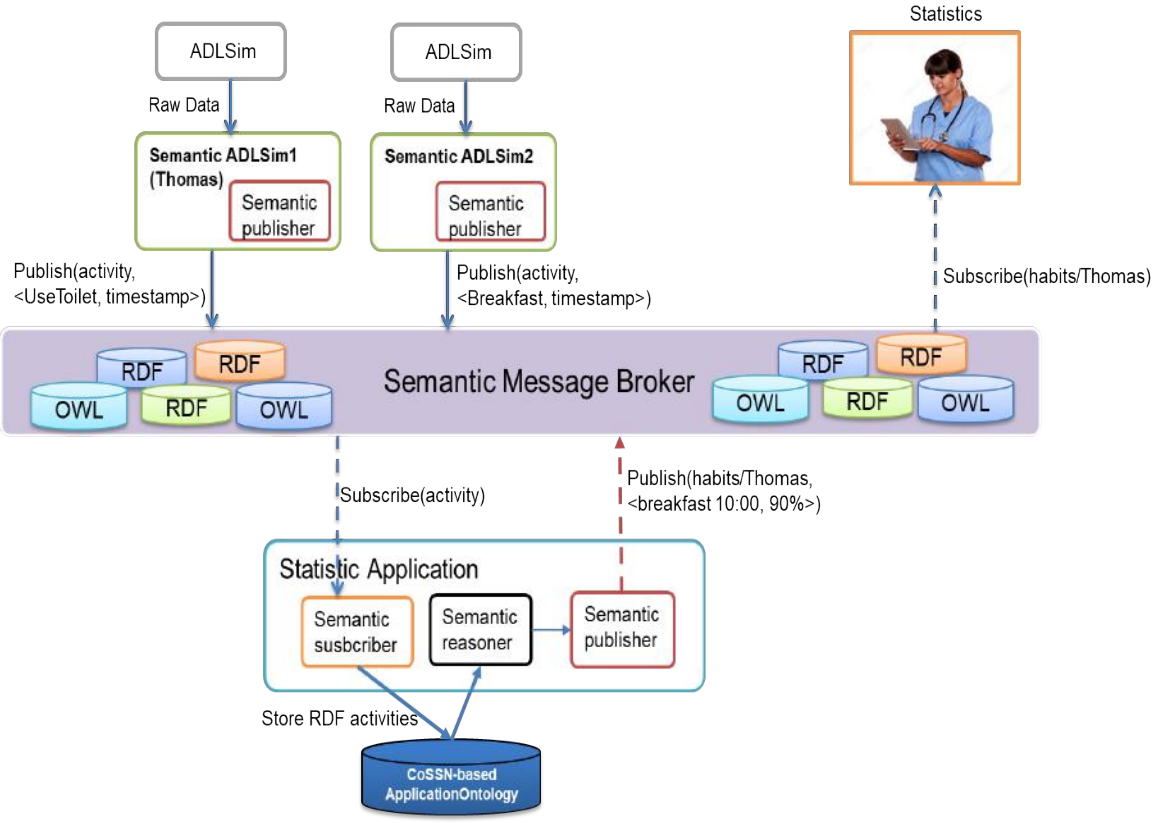


Figure 5.11 Semantic ADLSim component in SeMoM architecture

5.4.2 ADLSim Simulator Experimentation

In order to deploy the framework, the same technical choice as the deployment in the CHL has been made regarding the MQTT and RDF4J libraries. The main difference lies on replacing physical sensors with ADLSim as presented in the deployment diagram in Figure. 5.12. Two main components have been deployed: the semantic "ADLSim" and the "StatisticsApplication".

The "SemanticADLSim" component is responsible for generating the activities, and for wrapping them into RDF triple sensorOutput based on the CoSSN ontology and by means of RDF4J. It uses the MQTT publisher to send the semantic observations to the broker, for example, "publish(activity/Thomas, UseToilet)" is used to send the "UseToilet" activity of Thomas by specifying the topic as "ssn:property/ssn:featureOfInterest" described in CoSSN ontology.

The "StatisticsApplication" component uses MQTT subscriber and RDF4J to receive and manipulate the received semantic observations. The application uses the SPARQL reasoning technique to detect the habits of the person such as "90% of the days the person wakes up at 10:00". When the application receives the activities from several ADLSim instances, it analyzes publishes to the broker personalized profiles specific to each person, for example publish(habits/Thomas, "90% Breakfast at 10:00").

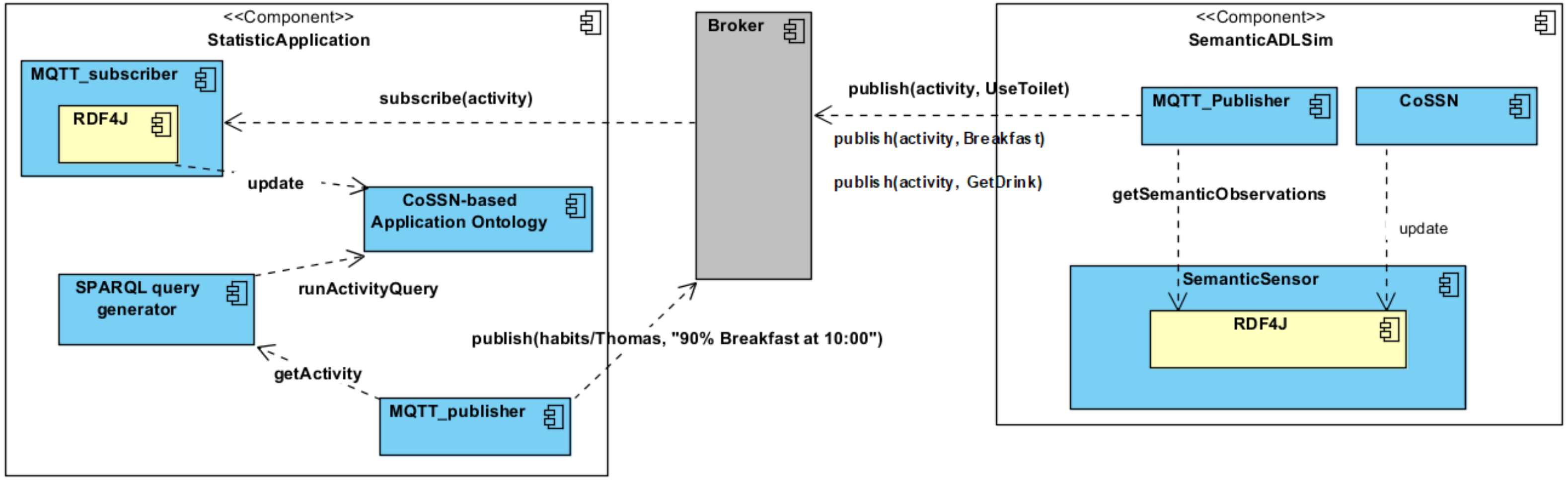


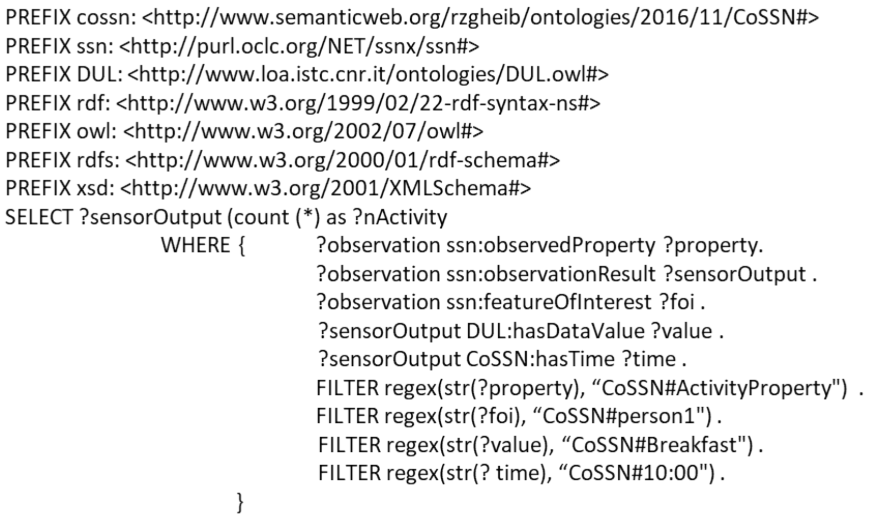
Figure 5.12 Deployment Diagram of ADLSim simulator with SeMoM

The objectives of this experimentation are twofold:

1. To implement SeMoM framework in a fixed scenario where data is reliable and replicable. In this case, we can ensure the reliability of CoSSN and the SPARQL query applied to a specific application.

2. In real experimental settings, it might be cumbersome or plainly impossible to test the scalability of a software architecture facing an increase in the number of generated sensor measurements. Using a simulator makes this task easy.

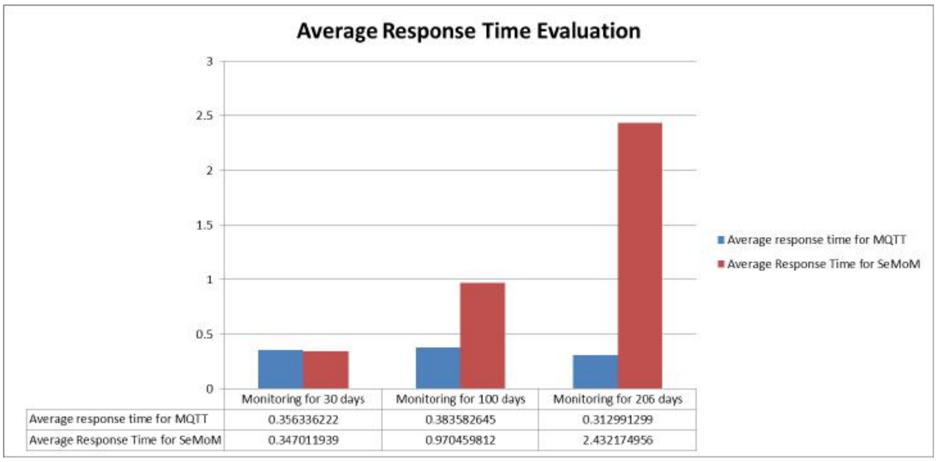
In order to evaluate the CoSSN ontology, we deployed one instance of ADLSim. We developed an IoT application that uses SPARQL query to detect the habits of one monitored person. We run the ADLSim and simulated the activities of a person for 30 days and we applied the SPARQL query presented in Figure.5.13. We have deduced that the result shows a high level of accuracy when data are simulated without lost messages.



```
PREFIX cossn: <http://www.semanticweb.org/rzgheib/ontologies/2016/11/CoSSN#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX DUL: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?sensorOutput (count (*) as ?nActivity
            WHERE {         ?observation ssn:observedProperty ?property.
                            ?observation ssn:observationResult ?sensorOutput .
                            ?observation ssn:featureOfInterest ?foi .
                            ?sensorOutput DUL:hasDataValue ?value .
                            ?sensorOutput CoSSN:hasTime ?time .
                            FILTER regex(str(?property), “CoSSN#ActivityProperty")  .
                            FILTER regex(str(?foi), “CoSSN#person1") .
                            FILTER regex(str(?value), “CoSSN#Breakfast") .
                            FILTER regex(str(? time), “CoSSN#10:00") .
                  }
```

Figure 5.13 Excerpt of SPARQL query rule used to detect the usual breakfast time for person1

From the other side, we evaluated the average response time of the framework when we use only MQTT broker and we compare it to when we send RDF triples over MQTT. This enables us to have an empirical evaluation of the overhead induced by adding the semantic layer, compared to sending raw sensor data instead. For that, we run the simulator for 3 different periods: 30 days, 100 days and 206 days. Results presented in Figure. 5.14 show that the difference of average response time between MQTT and SeMoM is not significant when we monitor a person for 30 days. However, this increases with the number of days and when we monitor a person for 206 days the average response time can reach 2.5s.



	Monitoring for 30 days	Monitoring for 100 days	Monitoring for 206 days
Average response time for MQTT	0.356336222	0.383582645	0.312991299
Average Response Time for SeMoM	0.347011939	0.970459812	2.432174956

Figure 5.14 Evaluation of the Response time performed by MQTT and SeMoM

5.5 Discussion

In this chapter we presented two deployments of our SeMoM framework. At the time of writing, the framework implemented in the CHL environment is able to handle sensors from different technologies. Moreover, the SPARQL query implemented for activity recognition is based on actions that occurred in the same time frame. For example, when the wardrobeButton is ON, we check at the same time if the person is not in bed and if he is present in the room. Likewise for the "taking shower" activity, when we get information from water sensor, we check the other measurements. However, more work should be done in order to consider sequences of actions respecting the order and the time of occurrence of these actions. To this end, our future work will consider other analytic technologies, such as Complex Event Processing [236] [237] wich appears promising.

It is important to notice that cognitive sensors need to store RDF triplets in an RDF triple store such as the CoSSN-based Ontology Application in order to run the SPARQL query. This process increases the execution time. It is not a problem for regular activities but it could be cumbersome for critical activities like falling down. CEP also appears well suited to tackle this performance problem, since this technique has been applied to time-critical problems such as intrusion detection [238].

Concerning the experimentations based on ADLSim, we chose to monitor one person. Further experimentations could be made such as monitoring several persons in several homes. Moreover, at the time of writing the performance evaluation has been limited to assess the impact of adding semantic level to raw data over MQTT.

From the other side, the role of developers in implementing an IoT system is hard and demanding work. In fact, developing solutions for the IoT requires collaboration, coordination, and connectivity for each component in the system, and throughout the system as a whole [239]. All devices must work together and be integrated with all other devices, and all devices must communicate and interact seamlessly with connected systems and infrastructures. An IoT programmer is responsible of offering a ready to use solution which consists on coupling deployment and manipulation of physical world with advances in machine learning and reasoning techniques.

The task of a programmer is not impossible but difficult and time consuming especially when it is related to domain-specific applications like healthcare. Domain specific applications require specific efforts and communication with experts in order to implement algorithms and decision making rules. Concretely, the complexity regarding the IoT developers is to define a topology of the multiple physical configurations, to connect several things based on different technologies such as bluetooth, zigbee, wifi, and the different libraries, frameworks and environments needed to implement a single IoT system. Furthermore, IoT developers must include data management providing the ability to capture and process tremendous amounts of data to predict new information based on the expert knowledge.
In this work, we provide a semantic middleware that facilitates the implementation of semantic IoT applications. In future work, this approach will be improved to make the task of the programmer more easier.

5.6 Conclusion

In this chapter, we have presented a human activity recognition case study to illustrate the SeMoM framework. We provided a full description of the proposed framework with all technologies under use. We detailed two experiments based on SeMoM: the first experiment based on real physical sensors while the other is based on the ADLSim simulator. We showed in the first experimentation that our framework can be deployed in real heterogeneous IoT environments. The second experimentation helped to evaluate the performance and reliability of the SeMoM when it is confronted with considerable amount of data generated by a simulator. We discussed the evaluation results of SeMoM and we showed that it is accurate in detecting the activities. However, adding semantics to the messages have impacts on the response time and CPU usage that increase when the number of exchanged data increases.

Finally, despite the importance of the semantic middleware combination for smart and flexible connection between IoT system components, another aspect of complexity related to the diversity of libraries to use and the complexity of programming environment has been observed in our work. In fact, the programmer has to use different libraries and has to integrate different technologies and mechanisms in a single environment (MQTT, RDF4J, Raspberry pi, arduino, etc.). Our final target in the future is to offer a ready to use platform based on the developed SeMoM framework.

Chapter 6

General Conclusion

"Happiness does not come from doing easy work but from the afterglow of satisfaction that comes after the achievement of a difficult task that demanded our best."

— *Theodore Isaac Rubin*

Contents

Today the IoT has become ubiquitous, has touched almost every domain, and is affecting human life in incredible ways. The IoT provides appropriate solutions for a wide range of applications where medical care and health care representing one of the most attractive ones. However, the number of heterogeneous medical sensors that can be deployed in an IoT system, the applications that can be created based on these sensors in an optimized way, and the flexible communication between sensors and applications present open challenges that researchers try to tackle in different projects. We hypothesized that adding semantic technologies over a middleware model could offer a solution with high interoperability and flexibility for IoT applications.

Although other approaches have been proposed for interoperability enhancement in IoT environments, they are often limited to a particular fixed model or technology. They address interoperability either at the technical level or at the semantic level. Concerning the solutions that offer a semantic approach, they are mostly based on ontologies for sensor data and metadata description. None of these ontologies propose a full description of the IoT system including virtual sensors and domain-related concepts. From a technical interoperability perspective, middleware solutions have been proposed in most of IoT platforms in the literature. Most solutions rely on the traditional SOA communication model, while new researches tends to investigate the MoM architectures that have shown effectiveness and remarkable performance in providing loosely coupled communication for IoT. There are few proposals that provide a full solution with a semantic middleware architecture. However, these solutions either rely on the SOA architecture paradigm (while we favor the MoM paradigm) or provide a specific and proprietary architecture, while we focus on open solutions.

This PhD thesis addresses these challenges by proposing SeMoM, a semantic middleware architecture for IoT applications that addresses the challenges from two perspectives: semantic and technical. SeMoM focuses on the main scientific problems we stated in chapter 1: (i) how to guarantee the interoperability of the IoT system with the presence of heterogeneous technologies of the possible connected sensors and applications, (ii) how to enable a loosely coupled communication between all IoT system components; and (iii) how to integrate a domain application requirements, for instance healthcare requirements, with respects to expert knowledge and patient comfort. The main contributions regarding the different aspects of the thesis are presented below.

6.1 Review of Contributions

Our main contribution consists in **providing a new semantic middleware approach based on the principles of weakly coupled software architecture driven by semantic data** as presented in chapter 2. In particular, we detail concrete contributions associated to this research work as follows:

- **We designed the CoSeOn Ontology, its syntax and semantics** in Chapter 3. This novel ontology has been designed following the ontology design pattern of an existent ontology, SSN. The SSN ontology has been designed and recommended by the W3C since 2006 to describe physical sensor data and metadata. Our objective with CoSeOn is to describe the cognitive process that drives detection of high level information such as activities and actions which is beyond the focus of SSN.
- **We defined the CoSSN ontology as a generic ontology integrating both SSN and CoSeOn ontologies**. The concepts and formalization of the ontology

is detailed in Chapter 3. Our objective of CoSSN is to provide a full semantic description of an IoT system including physical sensors, virtual sensors, expert knowledge and domain specifications. This new approach has been illustrated with a medical application of bedsore detection as a case study.

- **We proposed SeMoM, a semantic message oriented middleware architecture for IoT applications.** This approach has been fully detailed in chapter 4. SeMoM relies on a message oriented middleware (MoM) following the publish/subscribe paradigm for a loosely coupled communication between system components. These components can then exchange semantic observation data in a flexible manner. We showed how SeMoM uses CoSSN to allow the interoperability of sensor data so that the same information can potentially be provided by different sensors ensuring the redundancy of data. Furthermore, the proposed approach allows to define cognitive sensor components that are able to receive semantic data, apply rules through decision modules and infer higher-level semantic knowledge, such as human activities.

- **We have proposed a new solution for a medical application that detects the bedsore risk**, based on Braden scale as a risk assessment calculation. This case study has been described using the CoSSN ontology and realized on a MoM middleware. This use case has helped to clearly illustrate our design principles and its effectiveness in healthcare applications. Based on this case study, we have shown that our approach can be used for real life applications.

- **We proposed and implemented the SeMoM framework by deploying an ADL detection system in a real smart place** presented in Chapter 5. CoSSN ontology has been extended to application-specific ontology including ADL concepts description and a full description of the deployment of SeMoM has been also presented. Our objective in this implementation is to provide a proof of concepts of our approach. We have shown the benefit of using semantics in a Message Oriented Middleware to have an interoperable and scalable system in an IoT application.

- **We have provided another deployment of the SeMoM framework leveraging the ADLSim simulator.** The ADLSim developed by the DMMS team of the UiO university generates seven defined activities of a monitored person at home. The collaboration with the DMMS team led to implement ADLSim as a semantic publisher component in our SeMoM framework. Hence it was useful to generate simulated sensor data and create a statistical applications that could help to evaluate our framework in the future.

6.2 Future Work

The work presented in this thesis provides contributions that enable a flexible communication between IoT system components and sharing semantic messages between them. It constitutes a step forward building a semantic IoT platform for IT developers. However, we have identified the future related work to follow up, which in some cases we have started to explore. Based on that, we identified the following potential future research directions.

In this thesis we propose a framework that enables the development of semantic IoT healthcare applications. However, we plan on developing a reference implementation of our proposed middleware. Our goal is to provide a simple software API usable in various application domains and to alleviate the tasks of software developers. In light of this, we investigate some technologies that will be integral part of the proposed API: the OSGi and the Kura platform. From one hand, OSGi is a Java framework for developing and deploying modular software programs and libraries. OSGi has shown a good performance in management of physical sensors at the technical layer [240]. From the other hand, Eclipse foundation has launched in january 2017 the Kura platform [1]. It integrates MQTT communication protocol and facilities to implement and add IoT devices. However this platform does not integrate semantic descriptions for sensors and observations. We plan to base our software API on the Kura platforms and add semantic enhancements to it.

From applicative point of view, we plan to validate our bedsore application through controlled evaluations in a clinical environment, to assess its performance, reliability, safety and usability by the nursing staff. We are also planning to go further in the ADL detection application to detect the person's habits and behaviors over a long period of time. This improvement will allow us to analyze and study a new medical application such as the Alzheimer disease. We will investigate the application of alternative algorithms like Complex Event Processing (CEP) for real-time processing of activity detection in order to improve the real-time detection of emergencies. Examples of existing CEP systems are SQLstream, StreamInsight, EVAM, or Esper. Moreover, healthcare was a domain of choice for illustration and implementation. In the future, it is possible to investigate different domains such as transportation or manufacturing for

[1] http://www.eclipse.org/kura/

instance.

Finally, research conducted during this PhD thesis helped addressing challenges in different fields: IoT middleware architecture, semantic description and the healthcare application. The outcomes of our research work opened the way for exploring other related research problems for the various fields that we mentioned briefly in this section. Security & privacy: As we are working in a specific domain like healthcare, security of communication and privacy of information are very important. These two aspects have not been addressed in this research work but securing the SeMoM framework could be a research field in the future.

Cloud-based for scalability: The integration of cloud-based concepts would be a direction for the SeMoM platform for better scalability. Some commercial solutions have already proposed the publish/subscribe broker MQTT on the cloud such as CloudMQTT[2] and dioty [3].

[2] https://www.cloudmqtt.com/

[3] http://www.dioty.co/

Appendix A

Sample script of semantic sensor publishing semantic observations via MQTTPublisher

```java

package com.grovepi.mqtt.semantic.sensor;

import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.rdf4j.model.Model;
import com.grovepi.mqtt.connection.MqttPublisher;
import com.grovepi.physical.sensors.jgrove.groveButtonSensor;
import com.grovepi.semantic.enrichment.SemanticSensor;
import grovepi.Pin;

public class SemanticGroveButtonSensor extends Thread {
  public static void main(String[] args) throws Exception {
    groveButtonSensor button = new groveButtonSensor(Pin.DIGITAL_PIN_4);
    SemanticSensor ButtonSensor = new SemanticSensor("ButtonSensor", "wardrobeButtonPIN4", "wardrobeOpened","behaviour","");
    ButtonSensor.addSensorToOntology();
    MqttPublisher app = new MqttPublisher("ButtonClient-Pub");
    app.runClient();
    for(;;)
    if(button.buttonNotPressed()){
      ButtonSensor.addObservation(0, ButtonSensor.getDatetime());
      Model result = ButtonSensor.getSensorOutput();
      String resultat = result.toString();
      String res = resultat.substring(2, resultat.length()-2);
      MqttMessage msg = new MqttMessage(res.toString().getBytes());
      app.sendMessage("wardrobeOpened", msg.toString());
      Thread.sleep(1000);
    }
  }
}
```

Listing A.1 Semantic sensor example

Appendix B

Sample script of a virtual semantic sensor

```

public class DressingUp_App {
private static  boolean PUBLISHER = false;
private static  boolean SUBSCRIBER = true;


public static void main(String[] args) throws MqttException,
   InterruptedException, RDFParseException, RepositoryException,
   IOException {

VirtualSemanticSensor wearingClothes = new VirtualSemanticSensor(
   "DressingUpSensor",
"DressingUpSensor1", "Dressing","activity", "");
wearingClothes.addSensorToOntology();

MqttPublisherSubscriber app =  new MqttPublisherSubscriber("
   DressingUp-Client");
app.runClient();
String[] topics = new String[]{"personInBed","personUp","
   wardrobeOpened"};
ApplicationRulesQueries wearingClothesquery = new
   ApplicationRulesQueries();
String activity = "";

for (;;){
  if(SUBSCRIBER){
  app.subscribeTO(topics);
  Thread.sleep(20000);
  PUBLISHER = true;
  }

if(PUBLISHER){
SUBSCRIBER = false;

BindingSet res = wearingClothesquery.runQuery("dressing");
if (res != null ){
  activity = "Wearing his clothes at time: " + res.toString();
  wearingClothes.addObservation(activity);
  app.sendMessage("activity", res.toString());
  }
Thread.sleep(1000);
SUBSCRIBER = true;
}
}

}
}
```

Listing B.1 Virtual semantic sensor example

Bibliography

[1] Julien Mineraud, Oleksiy Mazhelis, Xiang Su, and Sasu Tarkoma. A gap analysis of internet-of-things platforms. *Computer Communications*, 89:5–16, 2016.

[2] In Lee and Kyoochun Lee. The Internet of Things (IoT): Applications investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, jul 2015.

[3] Rolf H Weber and Romana Weber. *Internet of things*, volume 12. Springer, 2010.

[4] Maciej Kranz. *Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry.* John Wiley & Sons, 2016.

[5] Iuliana Chiuchisan, Hariton-Nicolae Costin, and Oana Geman. Adopting the internet of things technologies in health care systems. In *Electrical and Power Engineering (EPE), 2014 International Conference and Exposition on*, pages 532–535. IEEE, 2014.

[6] Hon Prof John R Beard and David E Bloom. Towards a comprehensive public health response to population ageing. *Lancet (London, England)*, 385(9968):658, 2015.

[7] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590, 2013.

[8] Garner. Gartner says the Internet of Things will transform the data center. *Retrieved from http://www.gartner.com/newsroom/id/2684616*, March 19 2014.

[9] Charalampos Doukas and Ilias Maglogiannis. Bringing iot and cloud computing towards pervasive healthcare. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 922–926. IEEE, 2012.

[10] Ekaterina Balandina, Sergey Balandin, Yevgeni Koucheryavy, and Dmitry Mouromtsev. IoT Use Cases in Healthcare and Tourism. In *2015 IEEE 17th Conference on Business Informatics*. Institute of Electrical & Electronics Engineers (IEEE), jul 2015.

[11] Norbert Noury, Thierry Hervé, Vicent Rialle, Gilles Virone, Eric Mercier, Gilles Morey, Aldo Moro, and Thierry Porcheron. Monitoring behavior in home using a smart fall sensor and position sensors. In *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000*, pages 607–610. IEEE, 2000.

[12] Anthony Fleury, Michel Vacher, and Norbert Noury. Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. *IEEE transactions on information technology in biomedicine*, 14(2):274–283, 2010.

[13] Norbert Noury, Anthony Fleury, Pierre Rumeau, AK Bourke, GO Laighin, Vincent Rialle, and JE Lundy. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666. IEEE, 2007.

[14] Alexander Gluhak, Srdjan Krco, Michele Nati, Dennis Pfisterer, Nathalie Mitton, and Tahiry Razafindralambo. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, 49(11), 2011.

[15] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti, and Subhajit Dutta. Role of middleware for internet of things: A study. *International Journal of Computer Science and Engineering Survey*, 2(3):94–105, 2011.

[16] Sten Hanke, Christopher Mayer, Oliver Hoeftberger, Henriette Boos, Reiner Wichert, Mohammed-R. Tazari, Peter Wolf, and Francesco Furfari. universAAL – An Open and Consolidated AAL Platform. In *Ambient Assisted Living*, pages 127–140. Springer Nature, 2011.

[17] Christophe Lohr, Philippe Tanguy, and Jérôme Kerdreux. xAAL: A Distributed Infrastructure for Heterogeneous Ambient Devices. *Journal of Intelligent Systems*, 24(3), jan 2015.

[18] Nourhene Alaya, Sadok Ben Yahia, and Myriam Lamolle. Towards unveiling the ontology key features altering reasoner performances. *arXiv preprint arXiv:1509.08717*, 2015.

[19] Jaeho Kim and Jang-Won Lee. OpenIoT: An open service framework for the Internet of Things. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*. Institute of Electrical and Electronics Engineers (IEEE), mar 2014.

[20] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science Services and Agents on the World Wide Web*, 17:25–32, dec 2012.

[21] Derlis Gregor, S Toral, T Ariza, Federico Barrero, Raúl Gregor, Jorge Rodas, and Mario Arzamendia. A methodology for structured ontology construction applied to intelligent transportation systems. *Computer Standards & Interfaces*, 47:108–119, 2016.

[22] Tsung-Jung Wu, Lynn M Schriml, Qing-Rong Chen, Maureen Colbert, Daniel J Crichton, Richard Finney, Ying Hu, Warren A Kibbe, Heather Kincaid, Daoud Meerzaman, et al. Generating a focused view of disease ontology cancer terms for pan-cancer data integration and analysis. *Database*, 2015, 2015.

[23] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 345–354. ACM, 2013.

[24] Byung Mun Lee and Jinsong Ouyang. Intelligent healthcare service by using collaborations between IoT personal health devices. *blood pressure*, 10:11, 2014.

[25] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, sep 2012.

[26] Mahmoud Elkhodr, Seyed Shahrestani, and Hon Cheung. The internet of things: new interoperability, management and security challenges. *arXiv preprint arXiv:1604.04824*, 2016.

[27] Stephan Haller, Stamatis Karnouskos, and Christoph Schroth. The internet of things in an enterprise context. In *Future Internet Symposium*, pages 14–28. Springer, 2008.

[28] Dhananjay Singh, Gaurav Tripathi, and Antonio J. Jara. A survey of Internet-of-Things: Future vision architecture, challenges and services. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*. Institute of Electrical & Electronics Engineers (IEEE), mar 2014.

[29] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013.

[30] Jose M. Chaquet, Enrique J. Carmona, and Antonio Fernández-Caballero. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633–659, jun 2013.

[31] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, sep 2012.

[32] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, jun 2010.

[33] Leonardo Onofri, Paolo Soda, Mykola Pechenizkiy, and Giulio Iannello. A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Systems with Applications*, 63:97–111, nov 2016.

[34] Lukun Wang. Recognition of Human Activities Using Continuous Autoencoders with Wearable Sensors. *Sensors*, 16(2):189, feb 2016.

[35] Alessandra Moschetti, Laura Fiorini, Dario Esposito, Paolo Dario, and Filippo Cavallo. Recognition of Daily Gestures with Wearable Inertial Rings and Bracelets. *Sensors*, 16(8):1341, aug 2016.

[36] Oscar D. Lara and Miguel A. Labrador. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[37] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pages 1–10. VDE, 2010.

[38] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. A Survey of Online Activity Recognition Using Mobile Phones. *Sensors*, 15(1):2059–2085, jan 2015.

[39] Francesco Soldovieri and Gianluca Gennarelli. Exploitation of Ubiquitous Wi-Fi Devices as Building Blocks for Improvised Motion Detection Systems. *Sensors*, 16(3):307, feb 2016.

[40] Beichen Chen, Zhong Fan, and Fengming Cao. Activity Recognition Based on Streaming Sensor Data for Assisted Living in Smart Homes. In *2015 International Conference on Intelligent Environments*. Institute of Electrical and Electronics Engineers (IEEE), jul 2015.

[41] Narayanan C. Krishnan and Diane J. Cook. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10:138–154, feb 2014.

[42] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors. *Sensors*, 16(4):426, mar 2016.

[43] Zhen Li, Zhiqiang Wei, Lei Huang, Shugang Zhang, and Jie Nie. Hierarchical Activity Recognition Using Smart Watches and RGB-Depth Cameras. *Sensors*, 16(10):1713, oct 2016.

[44] Sara Amendola, Rossella Lodato, Sabina Manzari, Cecilia Occhiuzzi, and Gaetano Marrocco. Rfid technology for iot-based personal healthcare in smart spaces. *IEEE Internet of Things Journal*, 1(2):144–152, 2014.

[45] Nobuo Saito. Ecological home network: An overview. *Proceedings of the IEEE*, 101(11):2428–2435, 2013.

[46] Zhonghai Wang and Xiaoya Xu. Smart home m2m networks architecture. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*, pages 294–299. IEEE, 2013.

[47] Yao-lin Zhu, Rong Li, Xue-bin Liu, and Xu Jian. Wireless communication technology in family health monitoring system. In *Business Management and Electronic Information (BMEI), 2011 International Conference on*, volume 3, pages 64–67. IEEE, 2011.

[48] Young-jin Park and Hui-sup Cho. Transmission of ecg data with the patch-type ecg sensor system using bluetooth low energy. In *ICT Convergence (ICTC), 2013 International Conference on*, pages 289–294. IEEE, 2013.

[49] Martin Bor, John Edward Vidler, and Utz Roedig. Lora for the internet of things. 2016.

[50] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

[51] Inmaculada Plaza, Lourdes MartíN, Sergio Martin, and Carlos Medrano. Mobile applications in an aging society: Status and trends. *Journal of Systems and Software*, 84(11):1977–1988, 2011.

[52] Mari Carmen Domingo. An overview of the internet of things for people with disabilities. *Journal of Network and Computer Applications*, 35(2):584–596, 2012.

[53] María José Morón, Alejandro Gómez-Jaime, JR Luque, and Eduardo Casilari. Development and evaluation of a python telecare system based on a bluetooth body area network. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):629526, 2011.

[54] Moeen Hassanalieragh, Alex Page, Tolga Soyata, Gaurav Sharma, Mehmet Aktas, Gonzalo Mateos, Burak Kantarci, and Silvana Andreescu. Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges. In *Services Computing (SCC), 2015 IEEE International Conference on*, pages 285–292. IEEE, 2015.

[55] Gregorio López, Víctor Custodio, and José Ignacio Moreno. Lobin: E-textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1446–1458, 2010.

[56] Sten Hanke, Christopher Mayer, Oliver Hoeftberger, Henriette Boos, Reiner Wichert, Mohammed-R Tazari, Peter Wolf, and Francesco Furfari. universAAL–an open and consolidated AAL platform. In *Ambient assisted living*, pages 127–140. Springer, 2011.

[57] Andrew Sixsmith, Sonja Meuller, Felicitas Lull, Michael Klein, Ilse Bierhoff, Sarah Delaney, and Robert Savage. SOPRANO–an ambient assisted living system for supporting older people at home. In *International Conference on Smart Homes and Health Telematics*, pages 233–236. Springer, 2009.

[58] Majid A Al-Taee, Waleed Al-Nuaimy, Ali Al-Ataby, Zahra J Muhsin, and Suhail N Abood. Mobile health platform for diabetes management based on the internet-of-things. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*, pages 1–5. IEEE, 2015.

[59] Angelika Dohr, Robert Modre-Opsrian, Mario Drobics, Dieter Hayn, and Günter Schreier. The internet of things for ambient assisted living. In *Information technology: new generations (ITNG), 2010 seventh international conference on*, pages 804–809. Ieee, 2010.

[60] Maged N Kamel Boulos and Najeeb M Al-Shorbaji. On the internet of things, smart cities and the who healthy cities. *International journal of health geographics*, 13(1):10, 2014.

[61] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient intelligence in assisted living: enable elderly people to handle future interfaces. In *International Conference on Universal Access in Human-Computer Interaction*, pages 103–112. Springer, 2007.

[62] Michele Amoretti, Sergio Copelli, Folker Wientapper, Francesco Furfari, Stefano Lenzi, and Stefano Chessa. Sensor data fusion for activity monitoring in the persona ambient assisted living project. *Journal of Ambient Intelligence and Humanized Computing*, 4(1):67–84, 2013.

[63] Mukhtiar Memon, Stefan Rahr Wagner, Christian Fischer Pedersen, Femina Hassan Aysha Beevi, and Finn Overgaard Hansen. Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. *Sensors*, 14(3):4312–4341, 2014.

[64] Matthias Wagner, Benjamin Kuch, Carlos Cabrera, Peter Enoksson, and Arne Sieber. Android based body area network for the evaluation of medical parameters. In *Intelligent Solutions in Embedded Systems (WISES), 2012 Proceedings of the Tenth Workshop on*, pages 33–38. IEEE, 2012.

[65] Val Jones, Aart van Halteren, Nikolai Dokovsky, George Koprinkov, Jan Peuscher, Richard Bults, Dimitri Konstantas, Rainer Herzog, et al. Mobihealth: mobile services for health professionals. In *M-Health*, pages 237–246. Springer, 2006.

[66] Yuxi Liu and Guohui Zhou. Key technologies and applications of internet of things. In *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*, pages 197–200. IEEE, 2012.

[67] Juan Ye, Simon Dobson, and Susan McKeever. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1):36–66, feb 2012.

[68] Grégory Mermoud, Jean-Philippe Vasseur, and Sukrit Dasgupta. Hidden markov model based architecture to monitor network node activities and predict relevant periods, July 31 2013. US Patent App. 13/955,648.

[69] Peter JF Lucas, Linda C Van der Gaag, and Ameen Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial intelligence in medicine*, 30(3):201–214, 2004.

[70] Georgios Meditskos, Stamatia Dasiopoulou, and Ioannis Kompatsiaris. MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns. *Pervasive and Mobile Computing*, 25:104–124, jan 2016.

[71] Liming Chen, Chris D. Nugent, and Hui Wang. A Knowledge-Driven Approach to Activity Recognition in Smart Homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974, jun 2012.

[72] Liming Chen, Chris Nugent, and George Okeyo. An Ontology-Based Hybrid Approach to Activity Modeling for Smart Homes. *IEEE Transactions on Human-Machine Systems*, 44(1):92–105, feb 2014.

[73] Daniele Riboni and Claudio Bettini. OWL 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 7(3):379–395, jun 2011.

[74] Eric Prud, Andy Seaborne, et al. Sparql query language for rdf. 2006.

[75] Jean-Paul Calbimonte, Hoyoung Jeung, Oscar Corcho, and Karl Aberer. Semantic sensor data search in a large-scale federated sensor network. In *Proceedings of the 4th International Conference on Semantic Sensor Networks-Volume 839*, pages 23–38. CEUR-WS. org, 2011.

[76] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[77] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[78] Moumena A. Chaqfeh and Nader Mohamed. Challenges in middleware solutions for the internet of things. In *2012 International Conference on Collaboration Technologies and Systems (CTS)*. Institute of Electrical and Electronics Engineers (IEEE), may 2012.

[79] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, and Siobhan Clarke. Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 3(1):70–95, feb 2016.

[80] J Radhika and S Malarvizhi. Middleware approaches for wireless sensor networks: An overview. *International Journal of Computer Science Issues (IJCSI)*, 9(3):224–229, 2012.

[81] Vaskar Raychoudhury, Jiannong Cao, Mohan Kumar, and Daqiang Zhang. Middleware for pervasive computing: A survey. *Pervasive and Mobile Computing*, 9(2):177–200, 2013.

[82] Hitha Alex, Mohan Kumar, and Behrooz Shirazi. Midfusion: An adaptive middleware for information fusion in sensor network applications. *Information Fusion*, 9(3):332–343, 2008.

[83] Markus C Huebscher and Julie A McCann. Adaptive middleware for context-aware applications in smart-homes. In *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, pages 111–116. ACM, 2004.

[84] Wendi B Heinzelman, Amy L Murphy, Hervaldo S Carvalho, and Mark A Perillo. Middleware to support sensor network applications. *IEEE network*, 18(1):6–14, 2004.

[85] Marco Mamei and Franco Zambonelli. *Field-based coordination for pervasive multiagent systems.* Springer Science & Business Media, 2006.

[86] Ting Liu and Margaret Martonosi. Impala: A middleware system for managing autonomic, parallel sensor systems. In *ACM SIGPLAN Notices*, volume 38, pages 107–118. ACM, 2003.

[87] Porlin Kang, Cristian Borcea, Gang Xu, Akhilesh Saxena, Ulrich Kremer, and Liviu Iftode. Smart messages: A distributed computing platform for networks of embedded systems. *The Computer Journal*, 47(4):475–494, 2004.

[88] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(3):16, 2009.

[89] Igor L Marques, Jobson Ronan, and Nelson Souto Rosa. Tinyreef: a register-based virtual machine for wireless sensor networks. In *Sensors, 2009 IEEE*, pages 1423–1426. IEEE, 2009.

[90] Kirak Hong, Jiin Park, Sungho Kim, Taekhoon Kim, Hwangho Kim, Bernd Burgstaller, and Bernhard Scholz. Tinyvm: an energy-efficient execution infrastructure for sensor networks. *Software: Practice and Experience*, 42(10):1193–1209, 2012.

[91] Luca Mottola, Amy L Murphy, and Gian Pietro Picco. Pervasive games in a mote-enabled virtual world using tuple space middleware. In *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, page 29. ACM, 2006.

[92] Carlo Curino, Matteo Giani, Marco Giorgetta, Alessandro Giusti, Amy L Murphy, and Gian Pietro Picco. Tinylime: Bridging mobile and sensor networks through middleware. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 61–72. IEEE, 2005.

[93] Paolo Costa, Luca Mottola, Amy L Murphy, and Gian Pietro Picco. Teenylime: transiently shared tuple space middleware for wireless sensor networks. In *Proceedings of the international workshop on Middleware for sensor networks*, pages 43–48. ACM, 2006.

[94] Karl Aberer, Manfred Hauswirth, and Ali Salehi. A middleware for fast and flexible sensor network deployment. In *Proceedings of the 32nd international conference on Very large data bases*, pages 1199–1202. VLDB Endowment, 2006.

[95] Dominique Guinard, Vlad Trifa, Stamatis Karnouskos, Patrik Spiess, and Domnic Savio. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE transactions on Services Computing*, 3(3):223–235, 2010.

[96] Flávia C Delicato, Paulo F Pires, and Thais Batista. *Middleware solutions for the Internet of Things.* Springer, 2013.

[97] Nader Mohamed and Jameela Al-Jaroodi. Service-oriented middleware approaches for wireless sensor networks. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–9. IEEE, 2011.

[98] Pål Evensen and Hein Meling. Sensewrap: A service oriented middleware with sensor virtualization and self-configuration. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on*, pages 261–266. IEEE, 2009.

[99] Edgardo Avilés-López and J Antonio García-Macías. Tinysoa: a service-oriented architecture for wireless sensor networks. *Service Oriented Computing and Applications*, 3(2):99–108, 2009.

[100] Gaetano F Anastasi, Enrico Bini, Antonio Romano, and Giuseppe Lipari. A service-oriented architecture for qos configuration and management of wireless sensor networks. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8. IEEE, 2010.

[101] Charith Perera, Prem Prakash Jayaraman, Arkady Zaslavsky, Dimitrios Georgakopoulos, and Peter Christen. Mosden: An internet of things middleware for resource constrained mobile devices. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 1053–1062. IEEE, 2014.

[102] Ni Zhu, Tom Diethe, Massimo Camplani, Lili Tao, Alison Burrows, Niall Twomey, Dritan Kaleshi, Majid Mirmehdi, Peter Flach, and Ian Craddock. Bridging e-health and the internet of things: The sphere project. *IEEE Intelligent Systems*, 30(4):39–46, 2015.

[103] Maddy Janse, Peter Vink, and Nikolaos Georgantas. Amigo architecture: Service oriented architecture for intelligent future in-home networks. In *European Conference on Ambient Intelligence*, pages 371–378. Springer, 2007.

[104] Ehsan Ullah Warriach, Eirini Kaldeli, Alexander Lazovik, and Marco Aiello. An interplatform service-oriented middleware for the smart home. *International Journal of Smart Home*, 7(1):115–141, 2013.

[105] Anna Osello, Andrea Acquaviva, Chiara Aghemo, Laura Blaso, Daniele Dalmasso, David Erba, Giovanni Fracastoro, Damien Gondre, Marco Jahn, Enrico Macii, et al. Energy saving in existing buildings by an intelligent use of interoperable icts. *Energy efficiency*, 6(4):707–723, 2013.

[106] Alberto MC Souza and Jose RA Amazonas. A novel smart home application using an internet of things middleware. In *Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2013 European Conference on*, pages 1–7. VDE, 2013.

[107] Marco Bazzani, Davide Conzon, Andrea Scalera, Maurizio A Spirito, and Claudia Irene Trainito. Enabling the iot paradigm in e-health solutions through the virtus middleware. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1954–1959. IEEE, 2012.

[108] Mohammad Mehedi Hassan, Hanouf Saad Albakr, and Hmood Al-Dossari. A cloud-assisted internet of things framework for pervasive healthcare in smart city environment. In *Proceedings of the 1st International Workshop on Emerging Multimedia Applications and Services for Smart Cities*, pages 9–13. ACM, 2014.

[109] Juan M Corchado, Javier Bajo, Dante I Tapia, and Ajith Abraham. Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare. *IEEE transactions on information technology in biomedicine*, 14(2):234–240, 2010.

[110] Andreas K Triantafyllidis, Vassilis G Koutkias, Ioanna Chouvarda, and Nicos Maglaveras. A pervasive health system integrating patient monitoring, status logging, and social sharing. *IEEE journal of biomedical and health informatics*, 17(1):30–37, 2013.

[111] Sakari Junnila, Harri Kailanto, Juho Merilahti, Antti-Matti Vainio, Antti Vehkaoja, Mari Zakrzewski, and Jari Hyttinen. Wireless, multipurpose in-home health monitoring platform: two case trials. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):447–455, 2010.

[112] Filippo Palumbo, Paolo Barsocchi, Francesco Furfari, and Erina Ferro. Aal middleware infrastructure for green bed activity monitoring. *Journal of Sensors*, 2013, 2013.

[113] Rajesh Kannan Megalingam, Goutham Pocklassery, Vivek Jayakrishnan, Galla Mourya, and Athul Asokan Thulasi. Smartphone based continuous monitoring system for home-bound elders and patients. In *Communications and Signal Processing (ICCSP), 2014 International Conference on*, pages 1173–1177. IEEE, 2014.

[114] Antonio Coronato. Uranus: A middleware architecture for dependable aal and vital signs monitoring applications. *Sensors*, 12(3):3145–3161, 2012.

[115] Christian Seeger, Kristof Van Laerhoven, and Alejandro Buchmann. Myhealthassistant: An event-driven middleware for multiple medical applications on a smartphone-mediated body sensor network. *IEEE journal of biomedical and health informatics*, 19(2):752–760, 2015.

[116] Agustinus Borgy Waluyo, Song Ying, Isaac Pek, and Jian Kang Wu. Middleware for wireless medical body area network. In *Biomedical Circuits and Systems Conference, 2007. BIOCAS 2007. IEEE*, pages 183–186. IEEE, 2007.

[117] Ivan M Delamer and Jose L Martinez Lastra. Service-oriented architecture for distributed publish/subscribe middleware in electronics production. *IEEE Transactions on Industrial Informatics*, 2(4):281–294, 2006.

[118] Boualem Benatallah and Hamid Motahari Nezhad. Service oriented architecture: Overview and directions. *Advances in Software Engineering*, pages 116–130, 2008.

[119] Emiliano Miluzzo, Nicholas D Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B Eisenman, Xiao Zheng, and Andrew T Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350. ACM, 2008.

[120] Niels Brouwers and Koen Langendoen. Pogo, a middleware for mobile phone sensing. In *Proceedings of the 13th International Middleware Conference*, pages 21–40. Springer-Verlag New York, Inc., 2012.

[121] The pachube feed cloud service, https://www.xively.com/.

[122] Internet of things – thingspeak service, https://thingspeak.com/.

[123] idigi device cloud, https://www.digi.com/.

[124] Dominique Guinard, Vlad Trifa, Friedemann Mattern, and Erik Wilde. From the internet of things to the web of things: Resource-oriented architecture and best practices. *Architecting the Internet of things*, pages 97–129, 2011.

[125] Dominique Guinard and Vlad Trifa. Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*, volume 15, 2009.

[126] Dennis Pfisterer, Kay Romer, Daniel Bimschas, Oliver Kleine, Richard Mietz, Cuong Truong, Henning Hasemann, Alexander Kröller, Max Pagel, Manfred Hauswirth, et al. Spitfire: toward a semantic web of things. *IEEE Communications Magazine*, 49(11):40–48, 2011.

[127] Pratikkumar Desai, Amit Sheth, and Pramod Anantharam. Semantic gateway as a service architecture for iot interoperability. In *Mobile Services (MS), 2015 IEEE International Conference on*, pages 313–319. IEEE, 2015.

[128] M Botts. Ogc implementation specification 07-000: Opengis sensor model language (sensorml)-open geospatial consortium. Technical report, Tech. Rep, 2007.

[129] Craig Schlenoff, Tsai Hong, Connie Liu, Roger Eastman, and Sebti Foufou. A literature review of sensor ontologies for manufacturing applications. In *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*, pages 96–101. IEEE, 2013.

[130] Sarfraz Alam, Mohammad MR Chowdhury, and Josef Noll. Senaas: An event-driven sensor virtualization approach for internet of things cloud. In *Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on*, pages 1–6. IEEE, 2010.

[131] Rimel Bendadouche, Catherine Roussey, Gil De Sousa, Jean-Pierre Chanet, and K Mean Hou. Etat de l'art sur les ontologies de capteurs pour une intégration intelligente des données. *INFORSID 2012*, pages 89–104, 2012.

[132] Mike Botts, George Percivall, Carl Reed, and John Davidson. Ogc® sensor web enablement: Overview and high level architecture. *GeoSensor networks*, pages 175–190, 2008.

[133] Michael Compton, Cory Henson, Laurent Lefort, Holger Neuhaus, and Amit Sheth. A survey of the semantic specification of sensors. In *Proceedings of the 2nd International Conference on Semantic Sensor Networks-Volume 522*, pages 17–32. CEUR-WS. org, 2009.

[134] R Bendadouche, C Roussey, G De Sousa, J P Chanet, K M Hou, and K M Hou Extension. Extension of the Semantic Sensor Network Ontology for Wireless Sensor Networks : The Stimulus-WSNnode-Communication Pattern. *5th International Workshop on Semantic Sensor Networks in conjunction with the 11th International Semantic Web Conference (ISWC)*, (Nov 2012, Boston, United States), 2013.

[135] Sasikanth Avancha, Chintan Patel, and Anupam Joshi. Ontology-driven adaptive sensor networks. In *MobiQuitous*, volume 4, pages 194–202, 2004.

[136] David J Russomanno, Cartik R Kothari, and Omoju A Thomas. Building a sensor ontology: A practical approach leveraging iso and ogc models. In *IC-AI*, pages 637–643. Citeseer, 2005.

[137] Christopher J Matheus, David Tribble, Mieczyslaw M Kokar, Marion G Ceruti, and Scott C McGirr. Towards a formal pedigree ontology for level-one sensor fusion. Technical report, VERSATILE INFORMATION SYSTEMS INC FRAMINGHAM MA, 2006.

[138] Jeong-Hee Kim, Hoon Kwon, Do-Hyeun Kim, Ho-Young Kwak, and Sang-Joon Lee. Building a service-oriented ontology for wireless sensor networks. In *Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on*, pages 649–654. IEEE, 2008.

[139] Mohamad Eid, Ramiro Liscano, and Abdulmotaleb El Saddik. A universal ontology for sensor networks data. In *Computational Intelligence for Measurement Systems and Applications, 2007. CIMSA 2007. IEEE International Conference on*, pages 59–62. IEEE, 2007.

[140] Matt Calder, Robert A Morris, and Francesco Peri. Machine reasoning about anomalous sensor data. *Ecological Informatics*, 5(1):9–18, 2008.

[141] Kenneth J Witt, Jason Stanley, David Smithbauer, Dan Mandl, Vuong Ly, Al Underbrink, and Mike Metheny. Enabling sensor webs by utilizing swamo for autonomous operations. In *8th NASA Earth Science Technology Conference*, pages 263–270, 2008.

[142] Arthur Herzog, Daniel Jacobi, and Alejandro Buchmann. A3me-an agent-based middleware approach for mixed mode environments. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on*, pages 191–196. IEEE, 2008.

[143] Mario Gomez, Alun Preece, Matthew Johnson, Geeth De Mel, Wamberto Vasconcelos, Christopher Gibson, Amotz Bar-Noy, Konrad Borowiecki, Thomas La Porta, Diego Pizzocaro, et al. An ontology-centric approach to sensor-mission assignment. *Knowledge Engineering: Practice and Patterns*, pages 347–363, 2008.

[144] Luis Bermudez, Eric Delory, Tom O'Reilly, and Joaquin del Rio Fernandez. Ocean observing systems demystified. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–7. IEEE, 2009.

[145] Carlos Rueda, Luis Bermudez, and Janet Fredericks. The mmi ontology registry and repository: A portal for marine metadata interoperability. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–6. IEEE, 2009.

[146] Holger Neuhaus and Michael Compton. The semantic sensor network ontology. In *AGILE workshop on challenges in geospatial data harmonisation, Hannover, Germany*, pages 1–33, 2009.

[147] Alia Ibrahim, Francois Carrez, and Klaus Moessner. Spatio-temporal model for role assignment in wireless sensor networks. In *Wireless Conference (EW), Proceedings of the 2013 19th European*, pages 1–6. VDE, 2013.

[148] Yazid Benazzouz, Ioannis Parissis, et al. A fault fuzzy-ontology for large scale fault-tolerant wireless sensor networks. *Procedia Computer Science*, 35:203–212, 2014.

[149] Fernando Roda and Estanislao Musulin. An ontology-based framework to support intelligent data analysis of sensor measurements. *Expert Systems with Applications*, 41(17):7914–7926, 2014.

[150] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Iot-o, a core-domain iot ontology to represent connected devices networks. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 561–576. Springer, 2016.

[151] Suparna De, Payam Barnaghi, Martin Bauer, and Stefan Meissner. Service modelling for the internet of things. In *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, pages 949–955. IEEE, 2011.

[152] Tongrang Fan and Yanzhao Chen. A scheme of data management in the internet of things. In *Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on*, pages 110–114. IEEE, 2010.

[153] Christophe Lohr, Philippe Tanguy, and Jérôme Kerdreux. xaal: A distributed infrastructure for heterogeneous ambient devices. *Journal of intelligent systems*, 24(3):321–331, 2015.

[154] Xiang Su, Hao Zhang, Jukka Riekki, Ari Keränen, Jukka K Nurminen, and Libin Du. Connecting iot sensors to knowledge-based systems by transforming senml to rdf. *Procedia Computer Science*, 32:215–222, 2014.

[155] Amelie Gyrard, Christian Bonnet, and Karima Boudaoud. Enrich machine-to-machine data with semantic web technologies for cross-domain applications. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 559–564. IEEE, 2014.

[156] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud. Standardizing generic cross-domain applications in internet of things. In *Globecom Workshops (GC Wkshps), 2014*, pages 589–594. IEEE, 2014.

[157] Hayder I Hendi, Adeel Ahmad, Mourad Bouneffa, and Cyril Fonlupt. Ontology based reasoning for solving passenger train optimization problem. In *Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on*, pages 1–6. IEEE, 2016.

[158] Pedro HS Duarte, Luis F Faina, Lasaro J Camargos, Luciano B de Paula, and Rafael Pasquini. An architecture for monitoring and improving public transportation systems. In *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*, pages 871–878. IEEE, 2016.

[159] Heidi Scott. Energy ontologies: Wind, biomass, and fossil transportation. *Humanities*, 5(2):37, 2016.

[160] Clement Jonquet, Anne Toulet, Elizabeth Arnaud, Sophie Aubin, Esther Dzalé Yeumo, Vincent Emonet, Valeria Pesce, and Pierre Larmande. Agroportal: an open repository of ontologies and vocabularies for agriculture and nutrition data. In *GODAN Summit*, 2016.

[161] Rayner Alfred, Kim On Chin, Patricia Anthony, Phang Wai San, Tan Li Im, Leow Ching Leong, and Gan Kim Soon. Ontology-based query expansion for supporting information retrieval in agriculture. In *The 8th International Conference on Knowledge Management in Organizations*, pages 299–311. Springer, 2014.

[162] Siquan Hu, Haiou Wang, Chundong She, and Junfeng Wang. Agont: ontology for agriculture internet of things. In *International Conference on Computer and Computing Technologies in Agriculture*, pages 131–137. Springer, 2010.

[163] Warren A Kibbe, Cesar Arze, Victor Felix, Elvira Mitraka, Evan Bolton, Gang Fu, Christopher J Mungall, Janos X Binder, James Malone, Drashtti Vasant, et al. Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic acids research*, page gku1011, 2014.

[164] Kevin Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.

[165] Nelia Lasierra, A Alesanco, S Guillén, and José García. A three stage ontology-driven solution to provide personalized care to chronic patients at home. *Journal of biomedical informatics*, 46(3):516–529, 2013.

[166] Jaime Cantais, David Dominguez, Valeria Gigante, Loredana Laera, and Valentina Tamma. An example of food ontology for diabetes control. In *Proceedings of the International Semantic Web Conference 2005 workshop on Ontology Patterns for the Semantic Web*, pages 1–9, 2005.

[167] Jonghun Kim, Jaekwon Kim, Daesung Lee, and Kyung-Yong Chung. Ontology driven interactive healthcare with wearable sensors. *Multimedia Tools and Applications*, 71(2):827–841, 2014.

[168] Paul Brandt, Twan Basten, Sander Stuiik, Vinh Bui, Paul de Clercq, Luís Ferreira Pires, and Marten van Sinderen. Semantic interoperability in sensor applications making sense of sensor data. In *Computational Intelligence in Healthcare and e-health (CICARE), 2013 IEEE Symposium on*, pages 34–41. IEEE, 2013.

[169] Sefki Kolozali, Maria Bermudez-Edo, Daniel Puschmann, Frieder Ganz, and Payam Barnaghi. A knowledge-based approach for real-time iot data stream annotation and processing. In *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE*, pages 215–222. IEEE, 2014.

[170] Krzysztof Janowicz and Michael Compton. The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In *Proceedings of the 3rd International Conference on Semantic Sensor Networks-Volume 668*, pages 64–78. CEUR-WS. org, 2010.

[171] Christopher Alexander. *The timeless way of building*, volume 1. New York: Oxford University Press, 1979.

[172] Aldo Gangemi. Ontology design patterns for semantic web content. In *International semantic web conference*, pages 262–276. Springer, 2005.

[173] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernandez-Lopez. The neon methodology framework: a scenario-based methodology for ontology development. *Applied ontology*, 10(2):107–145, 2015.

[174] Antonio De Nicola and Michele Missikoff. A lightweight methodology for rapid ontology engineering. *Communications of the ACM*, 59(3):79–86, 2016.

[175] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 2010.

[176] Ping Wang and Qingjun Liu. *Biomedical Sensors and Measurement.* Springer Berlin Heidelberg, 2011.

[177] iCore. D3.1: Virtual object requirements and dependencies. Technical report, European Commission under the 7th Framework Programme, 2012.

[178] Michele Nitti, Virginia Pilloni, Giuseppe Colistra, and Luigi Atzori. The virtual object as a major element of the internet of things: a survey. *IEEE Communications Surveys & Tutorials*, 18(2):1228–1240, 2016.

[179] Kay Römer, Thomas Schoch, Friedemann Mattern, and Thomas Dübendorfer. Smart identification frameworks for ubiquitous computing applications. *Wireless Networks*, 10(6):689–700, 2004.

[180] Ryan W Thomas, Daniel H Friend, Luiz A Dasilva, and Allen B Mackenzie. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*, 44(12):51–57, 2006.

[181] Qusay Mahmoud. *Cognitive networks: towards self-aware networks*. John Wiley & Sons, 2007.

[182] Eva Wollenberg, David Edmunds, and Louise Buck. Using scenarios to make decisions about the future: anticipatory learning for the adaptive co-management of community forests. *Landscape and urban planning*, 47(1):65–77, 2000.

[183] Andrey Somov, Corentin Dupont, and Raffaele Giaffreda. Supporting smart-city mobility with cognitive internet of things. In *Future Network and Mobile Summit (FutureNetworkSummit), 2013*, pages 1–10. IEEE, 2013.

[184] Harriet Green-GM and Watson IoT. Cognitive iot: Making the internet of things deliver for all of us. 2015.

[185] Fadi M Al-Turjman. Information-centric sensor networks for cognitive iot: an overview. *Annals of Telecommunications*, pages 1–16, 2016.

[186] Katharina Siorpaes and Martin Hepp. myontology: The marriage of ontology engineering and collective intelligence. *Bridging the Gep between Semantic Web and Web*, 2:127–138, 2007.

[187] John G Webster. *Prevention of pressure sores: engineering and clinical aspects*. CRC Press, 1991.

[188] Leandro R Solis, Adrian Liggins, Richard RE Uwiera, Niek Poppe, Enid Pehowich, Peter Seres, Richard B Thompson, and Vivian K Mushahwar. Distribution of internal pressure around bony prominences: implications to deep tissue injury and effectiveness of intermittent electrical stimulation. *Annals of biomedical engineering*, 40(8):1740–1759, 2012.

[189] Panachit Kittipanya-ngam and Fusionopolis Way. Bed Detection for Monitoring System in Hospital Wards. *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 5887–5890, 2012.

[190] KS Jaichandar and Edgar Alonzo Martínez García. Intelli-sense bed patient movement sensing and anti-sweating system for bed sore prevention in a clinical environment. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1–5. IEEE, 2011.

[191] Samantha Mei-che Pang and Thomas Kwok-shing Wong. Predicting pressure sore risk with the Norton, Braden, and Waterlow scales in a Hong Kong rehabilitation hospital. *Nursing Research*, 47(3):147–153, 1998.

[192] Doreen Norton. Calculating the risk. reflections on the norton scale. *Advances in Skin & Wound Care*, 2(3):24–31, 1989.

[193] Judy Waterlow. Pressure sores: a risk assessment card. *Nursing times*, 81(48):49–55, 1984.

[194] Nancy Bergstrom, Barbara J Braden, Antoinette Laguzza, and Victoria Holman. The braden scale for predicting pressure sore risk. *Nursing research*, 36(4):205–210, 1987.

[195] Zhexuan Song, Alvaro A Cárdenas, and Ryusuke Masuoka. Semantic middleware for the internet of things. In *Internet of Things (IOT), 2010*, pages 1–8. IEEE, 2010.

[196] Peter Wolf, Andreas Schmidt, Javier Parada Otte, Michael Klein, Sebastian Rollwage, and year=2010 Birgitta König-Ries and Torsten Dettborn and Aygul Gabdulkhakova. openaal - the open source middleware for ambient-assisted living (aal).

[197] Vlasios Tsiatsis, Alexander Gluhak, Tim Bauge, Frederic Montagut, Jesus Bernat, Martin Bauer, Claudia Villalonga, Payam Barnaghi, and Srdjan Krco. The sensei real world internet architecture. 2010.

[198] Peter Kostelnik, Martin Sarnovsk, and Karol Furdik. The semantic middleware for networked embedded systems applied in the internet of things and services domain. *Scalable Computing: Practice and Experience*, 12(3):307–316, 2011.

[199] Mauro Caporuscio, Pierre-Guillaume Raverdy, and Valerie Issarny. ubisoap: A service-oriented middleware for ubiquitous networking. *IEEE Transactions on Services Computing*, 5(1):86–98, 2012.

[200] Iván Corredor, José F Martínez, Miguel S Familiar, and Lourdes López. Knowledge-aware and service-oriented middleware for deploying pervasive services. *Journal of Network and Computer Applications*, 35(2):562–576, 2012.

[201] Amira Ben Hamida, Fabio Kon, Nelson Lago, Apostolos Zarras, Dionysis Athanasopoulos, Dimitris Pilios, Panos Vassiliadis, Nikolaos Georgantas, Valérie Issarny, Georgios Mathioudakis, et al. Integrated choreos middleware-enabling large-scale, qos-aware adaptive choreographies. 2013.

[202] Jesús Rodríguez-Molina, José-Fernán Martínez, Pedro Castillejo, and Rubén de Diego. SMArc: a proposal for a smart, semantic middleware architecture focused on Smart City energy management. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[203] M. Ben Alaya, Y. Banouar, T. Monteil, C. Chassot, and K. Drira. OM2M: Extensible ETSI-compliant M2M Service Platform with Self-configuration Capability. *Procedia Computer Science*, 32:1079–1086, 2014.

[204] Grégory Nain. *EnTiMid: Un modèle de composants pour intégrer des objets communicants dans des applications à base de services.* PhD thesis, Université Rennes 1, 2011.

[205] Kalil A Bispo, Nelson S Rosa, and Paulo RF Cunha. Sitrus: Semantic infrastructure for wireless sensor networks. *Sensors*, 15(11):27436–27469, 2015.

[206] Julien Mineraud, Oleksiy Mazhelis, Xiang Su, and Sasu Tarkoma. Contemporary internet of things platforms. *arXiv preprint arXiv:1501.07438*, 2015.

[207] Oleksiy Mazhelis and Pasi Tyrvainen. A framework for evaluating internet-of-things platforms: Application provider viewpoint. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 147–152. IEEE, 2014.

[208] Juan Luis Pérez, Álvaro Villalba, David Carrera, Iker Larizgoitia, and Vlad Trifa. The compose api for the internet of things. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 971–976. ACM, 2014.

[209] Berta Carballido Villaverde, Rodolfo de Paz Alberola, Antonio J Jara, Szymon Fedor, Sajal K Das, and Dirk Pesch. Service discovery protocols for constrained machine-to-machine communications. *IEEE Communications Surveys and Tutorials*, 16(1):41–60, 2014.

[210] Massimo Villari, Antonio Celesti, Maria Fazio, and Antonio Puliafito. Alljoyn lambda: An architecture for the management of smart environments in iot. In *Smart Computing Workshops (SMARTCOMP Workshops), 2014 International Conference on*, pages 9–14. IEEE, 2014.

[211] T Jyothi. Open Source Middleware for Internet of Things. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 4(11):71–75, 2016.

[212] Juan Luis Pérez and David Carrera. Performance characterization of the serviotic y api: an iot-as-a-service data management platform. In *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on*, pages 62–71. IEEE, 2015.

[213] Perdix iot platform, https://www.ge.com/digital/predix/industries.

[214] Ibm watson platform, https://www.ibm.com/watson/.

[215] Amazon aws, https://aws.amazon.com/.

[216] Azure iot suite, https://azure.microsoft.com/en-us/suites/iot-suite/.

[217] Tiziana Catarci, Claudio Di Ciccio, Vincenzo Forte, Ettore Iacomussi, Massimo Mecella, Giuseppe Santucci, and Giuseppe Tino. Service composition and advanced user interfaces in the home of tomorrow: The sm4all approach. In *International Conference on Ambient Media and Systems*, pages 12–19. Springer, 2011.

[218] Maddy D Janse. Amigo Final Report. pages 1–42, September 2008.

[219] Mirko Presser, Payam M Barnaghi, Markus Eurich, and Claudia Villalonga. The sensei project: Integrating the physical world with the digital world of the network of the future. *IEEE Communications Magazine*, 47(4):1–4, 2009.

[220] Vinita Shirley and Rithvik Pamidi. Web of things.

[221] Ali Larab, Emmanuel Conchon, Rémi Bastide, and Nicolas Singer. A sustainable software architecture for home care monitoring applications. In *Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on*, pages 1–6. IEEE, 2012.

[222] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.

[223] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. *Ambient intelligence*, 35:115–148, 2005.

[224] Qing Cao, Tarek Abdelzaher, John Stankovic, and Tian He. The liteos operating system: Towards unix-like abstractions for wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference On*, pages 233–244. IEEE, 2008.

[225] Emmanuel Baccelli, Oliver Hahm, Mesut Gunes, Matthias Wahlisch, and Thomas C Schmidt. Riot os: Towards an os for the internet of things. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 79–80. IEEE, 2013.

[226] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Harald Sundmaeker, Markus Eisenhauer, Klaus Moessner, Franck Le Gall, and Philippe Cousin. Internet of things strategic research and innovation agenda. *RIVER PUBLISHERS SERIES IN COMMUNICATIONS*, page 7, 2013.

[227] Dave Locke. Mqtt v3. 1 protocol specification. Technical report, Technical Report, International Business Machines Corporation (IBM) and Eurotech, 2010, 42p, 2010.

[228] Walter Colitti, Kris Steenhaut, Niccolo De Caro, Bogdan Buta, and Virgil Dobrota. Rest enabled wireless sensor networks for seamless integration with web applications. In *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, pages 867–872. IEEE, 2011.

[229] Peter Peniak and Maria Franekova. Open communication protocols for integration of embedded systems within industry 4. In *Applied Electronics (AE), 2015 International Conference on*, pages 181–184. IEEE, 2015.

[230] C Bormann, Z Shelby, and K Hartke. Constrained application protocol (coap), draft-ietf-core-coap-18, 2013.

[231] Peter Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. 2011.

[232] Advanced Open Standard for the Information Society (OASIS). Oasis advanced message queuing protocol (amqp) version 1.0, 2012.

[233] Hung Keng Pung, Tao Gu, Wenwei Xue, Paulito P Palmes, Jian Zhu, Wen Long Ng, Chee Weng Tang, and Nguyen Hoang Chung. Context-aware middleware for pervasive elderly homecare. *IEEE Journal on Selected Areas in Communications*, 27(4), 2009.

[234] James Y Xu, Yan Wang, Mick Barrett, Bruce Dobkin, Greg J Pottie, and William J Kaiser. Personalized multilayer daily life profiling through context enabled activity classification and motion reconstruction: An integrated system approach. *IEEE journal of biomedical and health informatics*, 20(1):177–188, 2016.

[235] Andrea Gaggioli, Giovanni Pioggia, Gennaro Tartarisco, Giovanni Baldus, Daniele Corda, Pietro Cipresso, and Giuseppe Riva. A mobile data collection platform for mental health research. *Personal and Ubiquitous Computing*, 17(2):241–251, 2013.

[236] Tawseef Naqishbandi, C Imthyaz Sheriff, and S Qazi. Big data, cep and iot: redefining holistic healthcare information systems and analytics. *Int J Eng Res and Technol*, 4(1):1–6, 2015.

[237] Ching Yu Chen, Jui Hsi Fu, Ping-Feng Wang, Emery Jou, and Ming-Whei Feng. Complex event processing for the internet of things and its applications. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 1144–1149. IEEE, 2014.

[238] Shang-Nan Yin, Ho-Seok Kang, Zhi-Guo Chen, and Sung-Ryul Kim. Intrusion detection system based on complex event processing in rfid middleware. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pages 125–129. ACM, 2016.

[239] Rajeev Alur, Emery Berger, Ann W Drobnis, Limor Fix, Kevin Fu, Gregory D Hager, Daniel Lopresti, Klara Nahrstedt, Elizabeth Mynatt, Shwetak Patel, et al. Systems computing challenges in the internet of things. *arXiv preprint arXiv:1604.02980*, 2016.

[240] Martin Stusek, Jiri Hosek, Dominik Kovac, Pavel Masek, Petr Cika, Jan Masek, and Franz Kröpfl. Performance analysis of the osgi-based iot frameworks on restricted devices as enablers for connected-home. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2015 7th International Congress on*, pages 178–183. IEEE, 2015.