Designing cross-domain semantic Web of things applications

Amélie Gyrard

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2015 ENST 0018

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique »

présentée et soutenue publiquement par

Amélie GYRARD

24 Avril 2015

Concevoir des applications

Internet des Objets Sémantiques inter-domaine

Directeur de thèse : **Christian BONNET**
Co-encadrement de la thèse : **Karima BOUDAOUD**

Jury

M. Bruno MARTIN, Professeur, Université Nice Sophia-Antipolis	Président
M. Jérôme EUZENAT, Directeur de Recherches, INRIA	Rapporteur
M. Oscar CORCHO, Professeur associé, Universidad Politécnica de Madrid	Rapporteur
M. Payam BARNAGHI, Assistant Professeur, University of Surrey	Examinateur
M. Claude HARY, Com4Innov	Examinateur
M. Philippe BADIA, Com4Innov	Invité
M. Christian BONNET, Professeur, Eurecom	Directeur de Thèse
Mme. Karima BOUDAOUD, Maître de Conférences, Université Nice Sophia-Antipolis	Directrice de Thèse

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

THÈSE

Designing Cross-Domain Semantic Web of Things Applications

Amélie Gyrard

April 2015

A doctoral dissertation submitted to:

TELECOM ParisTech

In Partial Fulfillment of the Requirements for the Degree of:

Doctor of Computer Science

Specialty : Internet of Things and Semantic Web

Thesis Supervisor: **Prof. Christian Bonnet**

Thesis Co-Supervisor: **Dr. Karima Boudaoud**

Jury:

President:

Prof. Bruno Martin - Université Nice Sophia-Antipolis, Sophia Antipolis - France

Reviewers:

Dr. Jérôme Euzenat - INRIA, Grenoble - France

Prof. Oscar Corcho - Universidad Politécnica de Madrid, Madrid - Spain

Examiners:

Dr. Payam Barnaghi - University of Surrey, Guildford - United Kingdom

Claude Hary - Com4Innov, Sophia Antipolis - France

Invited:

Philippe Badia - Com4Innov, Sophia Antipolis - France

Abstract

According to Cisco's predictions[1], there will be more than 50 billions of devices connected to the Internet by 2020. The devices and produced data are mainly exploited to build domain-specific Internet of Things (IoT) applications. From a data-centric perspective, these applications are not interoperable with each other. To assist users or even machines in building promising inter-domain IoT applications, main challenges are to exploit, reuse, interpret and combine sensor data. To overcome interoperability issues, we designed the Machine-to-Machine Measurement (M3) framework consisting in: (1) generating templates to easily build Semantic Web of Things applications, (2) semantically annotating IoT data to infer high-level knowledge by reusing as much as possible the domain knowledge expertise, and (3) a semantic-based security application to assist users in designing secure IoT applications. Regarding the reasoning part, stemming from the 'Linked Open Data', we propose an innovative idea called the 'Linked Open Rules' to easily share and reuse rules to infer high-level abstractions from sensor data. The M3 framework has been suggested to standardizations and working groups such as ETSI M2M, oneM2M, W3C SSN ontology and W3C Web of Things. Proof-of-concepts of the flexible M3 framework have been developed on the cloud (`http://www.sensormeasurement.appspot.com/`) and embedded on Android-based constrained devices.

[1] `http://share.cisco.com/internet-of-things.html`

French Abstract

Selon les prévisions de Cisco[2], il y aura plus de 50 milliards d'appareils connectés à Internet d'ici 2020. Les appareils et les données produites sont principalement exploitées pour construire des applications 'Internet des Objets' (IdO). D'un point de vue des données, ces applications ne sont pas interopérables les unes avec les autres. Pour aider les utilisateurs ou même les machines à construire des applications 'Internet des Objets' interdomaines innovantes, les principaux défis sont l'exploitation, la réutilisation, l'interprétation et la combinaison des données produites par les capteurs. Pour surmonter les problèmes d'interopérabilité, nous avons conçu le système Machine-to-Machine Measurement (M3) consistant à : (1) enrichir les données de capteurs avec les technologies du web sémantique pour décrire explicitement leur sens selon le contexte, (2) interpréter les données des capteurs pour en déduire des connaissances supplémentaires en réutilisant autant que possible la connaissance du domaine définie par des experts, et (3) une base de connaissances de sécurité pour assurer la 'sécurité dès la conception' lors de la construction des applications IdO. Concernant la partie raisonnement, inspiré par le 'Web de données', nous proposons une idée novatrice appelée le 'Web de règles' afin de partager et réutiliser facilement les règles pour interpréter les données de capteurs. Le système M3 a été suggéré à des normalisations et groupes de travail tels que l'ETSI M2M, oneM2M, W3C SSN et W3C Web of Things. Une preuve de concept de M3 a été implementée et est disponible sur le web (`http://www.sensormeasurement.appspot.com/`) et a été embarqué dans des appareils Android tels que les tablettes ou les téléphones mobiles.

[2] `http://share.cisco.com/internet-of-things.html`

Acknowledgments

Accomplishing this research has been an exciting journey with ups and downs. In the end, I realize that I have learned a lot, found my vocation and met great people around the world.

First of all, I would like to express my heartfelt gratitude to my supervisors, Christian Bonnet and Karima Boudaoud, who patiently guided me in my research and being here when I needed them. They put myself on the research directions with keywords ('semantics on data', 'rules', 'security & ontology', 'horizontal applications' and 'templates') which lead to my major contributions. Further, they helped me a lot emphasize my work. I am grateful for their kindness to let me work on research fields and scenarios that inspired me and let me present my work in conferences around the world (Rio de Janeiro, Seoul, Taipei, Riva Del Garda, etc.). I am grateful to the members of my thesis committee for their time and effort to evaluate this work. Thanks to their precious feedback and those from reviewers from conferences, I emphasize much better my work. I also say thanks to the project Com4Innov Platform of Pole SCS[3] and DataTweet[4] (ANR-13-INFR-0008) for funding my thesis.

I would like to thank semantic web experts who guided me throughout my thesis with their knowledge and expertise, particularly for teaching me the semantic web best practices and their precious feedback. I am thinking more specifically of Ghislain Atemezing, Payam Barnaghi, Raphael Troncy, Bernard Vatant, Fabien Gandon and Oliver Corby. Their help played an important role in my PhD and career. I would like to express my appreciation to Fabien Gandon who transmitted to me his passion for Semantic Web, and to Jean Marie Rifflet for security. Thanks to Soumya Kanti Datta for assisting me in improving and adapting the M3 framework to Android-powered devices.

I am also indebted to my lovely mother, she learned me to overcome the difficulties of life and finally enjoy the life much better. She also encouraged me every time I have doubted. I would also like to acknowledge my incredible companion Guillaume for every moment spent together. I have really enjoyed our mutual philosophical and witty discussions and our mutual concurrence to surpass ourselves. I would also like to acknowledge all martial art teachers who transmit me precious skills such as perseverance, combativeness, striving for perfection, daring, courage and body-mind connections for boosting brain power. A special thanks goes to my entire family, friends and colleagues (Ghislain, Jose, Guiseppe, Julien, Esther, Etienne, Jean, Fx, Ying, Anne-Claire, Claire-Astrid, Charlotte, etc.) for being there, listening to my worries, for their advice, feedback, questions and sharing the joy and laughs with me. I would also like to express my deep admiration for people who believe in me.

Finally, I would like to thank all authors of research articles that inspired me. I could not reference all papers that I read, so sorry in advance if I forget to reference your work even if it inspired me or if I wanted to reuse the ontologies and rules that you designed.

[3] http://www.pole-scs.org/

[4] http://www.agence-nationale-recherche.fr/?Projet=ANR-13-INFR-0008

[To all of you, simply thanks for everything you did.]

[To the quantum universe, the geniuses and the angels. They protect me, guide me and inspire me everyday.]

Contents

List of Figures

List of Tables

Part I

Introduction & State of the Art

In this first part, we introduce the thesis in Chapter 1 and the state of the art related to Semantic Web of Things in Chapter 2.

Chapter 1

Introduction

"You've got to find what you love. Have the courage to follow your heart and intuition."

Steve Jobs

1.1 Motivation

In recent years, we have been witnessing a growing number of sensors embedded in smart devices (e.g., mobile phones, smart watches or smart glasses) or everyday objects. Applications exploiting sensors and producing data are more and more popular. Domotic and smart healthcare are increasingly present in our everyday life. For example, Hapifork[1] tracks your eating habits. Oral-B[2] and Kolibree[3] connected toothbrush control dental hygiene. Mother[4] checks if you walk enough to stay fit, reminds medication, monitors the quality of your sleep, etc. The Apple HealthKit[5] tracks fitness, nutrition and sleep, etc. SFR Connected Homes[6] includes connected thermostat or lighting control. Sensors are deployed in smart farms and gardens too. Edyn[7] and Botanicalls[8] send alerts when plants need to be watered. Google Self-Driving Cars[9] are already allowed on public roads in Nevada or California. The smart devices are more and more connected to Internet and data is sent to the Web to build 'Internet of Things' (IoT) or 'Web of Things' applications. According to Cisco's predictions[10], there will be more than 50 billions of devices connected to the Internet by 2020. Due to the enormous quantity of sensor data produced, there is a real need to interpret these data and build interoperable IoT applications.

[1] http://www.hapi.com/product/hapifork
[2] http://connectedtoothbrush.com/
[3] http://www.kolibree.com/
[4] https://sen.se/store/mother/
[5] http://goo.gl/n2V42g
[6] http://connected-objects.fr/2014/05/sfr-home-box-domotique/
[7] https://www.kickstarter.com/projects/edyn/edyn-welcome-to-the-connected-garden
[8] http://www.botanicalls.com/
[9] http://en.wikipedia.org/wiki/Google_driverless_carandBMWconnectedcars
[10] http://share.cisco.com/internet-of-things.html

1.2 Problem

A first challenging problem is that devices are not interoperable with each other since their data is based on proprietary formats, they do not use common terms or vocabulary to describe **interoperable IoT data**. There is a similar issue with applications since they are based on proprietary protocols. One way to make them interoperable would be a common protocol used by all devices. Another solution would be to work on the interoperability of this data, since these devices are already deployed and data is already produced. Exploiting, combining and enriching this sensor data to build smarter interoperable applications is becoming a real challenge. The growing trend "Linked Open Data"[11] encourages to share the data on the web, including sensor data. To assist users or even machines in interpreting and combining these sensor data, there is a real need to explicitly describe sensor measurements according to the context, in a unified way and being understandable by machines. For instance, a temperature measurement has not the same meaning according to the context (room temperature, body temperature, water temperature or external temperature) and the machine will not infer the same knowledge (fever deduced with body temperature, abnormal temperature for a room temperature). We also need to deal with implicit units (e.g., Fahrenheit, Celsius, Kelvin).

The second biggest challenge is **combining domain-specific applications** and data together to create innovative cross-domain IoT applications. Existing applications are specific to one domain such as smart home, smart healthcare, transportation, smart garden, etc. Figure 1.1 shows examples of innovative cross-domain IoT applications: (1) suggesting food according to the weather forecasting, (2) suggesting home remedies according to health measurements, and (3) suggesting safety equipments in a smart car according to the weather, etc. Future smart fridges will enable to purchase groceries online. In case of RFID tags embedded in food, it will be easy to recommend the menu for dinner or automatically order essential ingredients [Xie et al., 2013] [Gu and Wang, 2009]. If you are an athlete, the smart fridge will recommend you the perfect diet [Tumnark et al., 2013] in case of compatibility with Apple Nike shoes[12]. Finally, Google's car could automatically stop you to the grocery store to grab the missing ingredients.

One of the most important challenge for the Internet of Things would be to assist developers in designing and developing interoperable inter-domain IoT applications.

Finally, **security issues should be considered when combining IoT data or designing IoT applications**. For instance, health data is more sensitive than weather data and needs to be secured.

1.3 Our Approach

In this thesis, we focus on the interoperability of sensor data to build promising and interoperable domain-specific or cross-domain IoT applications. To deal with this challenge,

[11] `http://linkeddata.org/`

[12] `https://www.apple.com/fr/ipod/nike/`

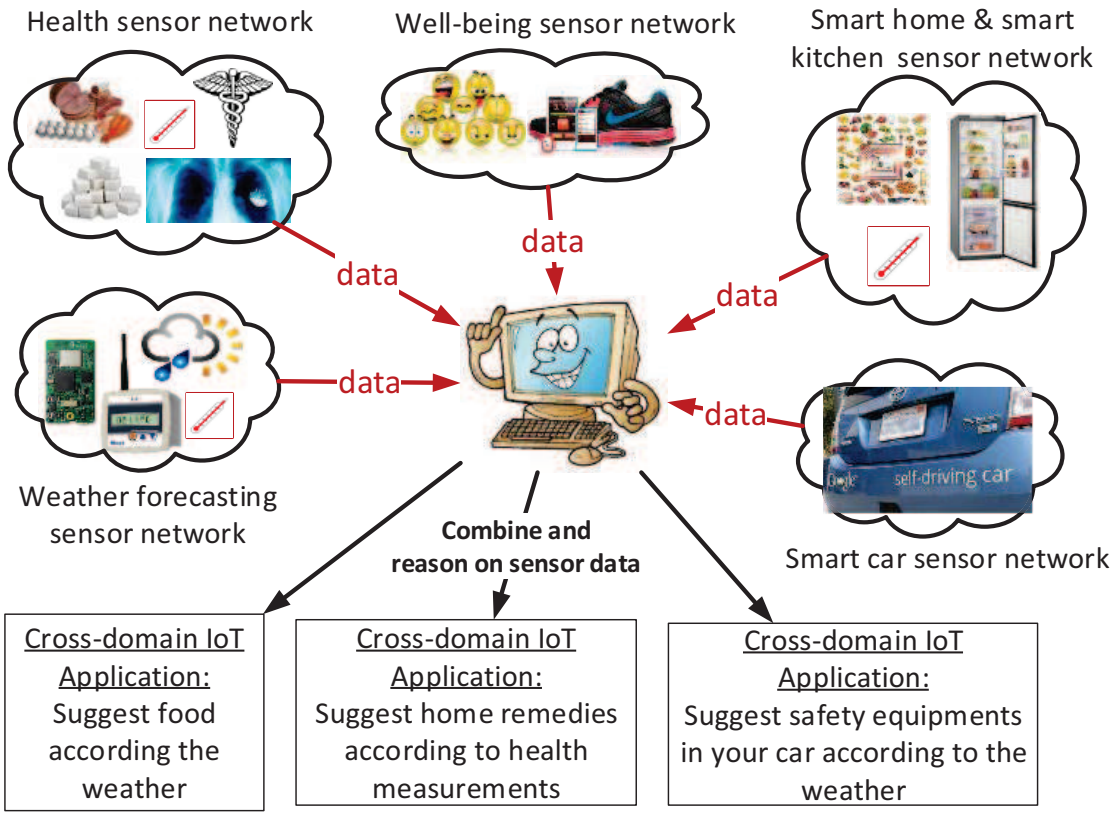

Figure 1.1: Combining domains to build promising IoT applications

we exploit semantic web technologies [Berners-Lee et al., 2001] for several reasons. Firstly, semantics enables an explicit description of the meaning of sensor data in a structured way, so that machines could understand it. Secondly, it facilitates interoperability for data integration since heterogeneous IoT data is converted according to the same vocabulary. Thirdly, semantic reasoning engines can be easily employed to deduce high-level abstractions from sensor data. Fourthly, context-awareness could be implemented using semantic reasoning. Finally, in theory, semantics eases the knowledge sharing and reuse of domain knowledge expertise which should avoid the reinvention of the wheel. Indeed, each time a new domain-specific vocabulary is defined.

Semantic web technologies are becoming very popular and are adopted by companies such as Google and Yahoo. Web sites use Schema.org[13] to enhance the research results. Schema.org is a set of vocabularies called ontologies [Gruber, 1993] to describe data on the Web in an unified way such as Person, Organization, etc. In Figure 1.2, on the right the search engine recognizes that Steve Jobs is a Person, his spouse is Lauren Powell Jobs, etc. Google introduces the idea of the knowledge graph to connect and structure the data with

[13] http://schema.org/

each other. Moreover, 'Linked data' is more and more popular to share and reuse data to build and enhance rich web applications with little effort [Tu, 2009]. We would like to design a similar approach to structure sensor data and analyze it to build cross-domain IoT applications.

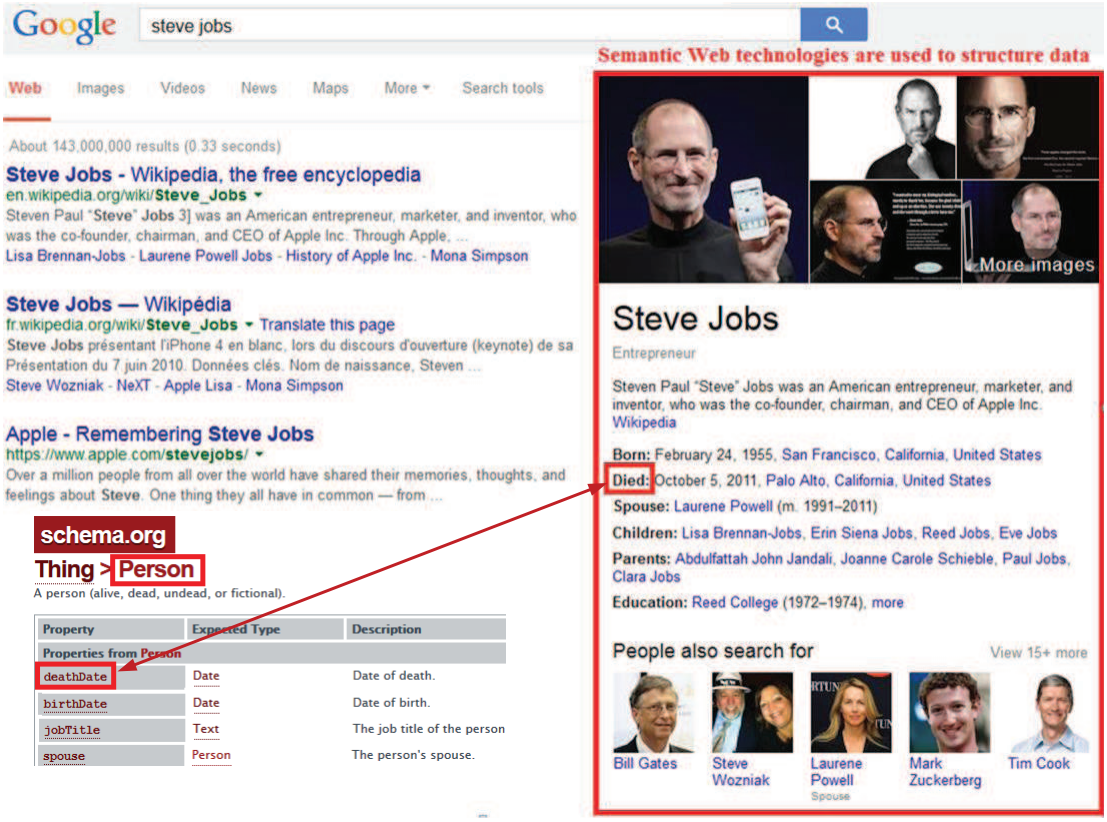

Figure 1.2: Semantic Web technologies used by Google to structure data on the Web

Further, according to Barnaghi et al., semantics is required at different levels in IoT, it can be used to: (1) describe things and data, (2) reuse domain knowledge, (3) interpret IoT data, (4) provide smarter applications, and (5) provide security [Barnaghi et al., 2012b].

In this thesis, we address the following challenges:

- Generating interoperable cross-domain semantic-based IoT applications. This process should be flexible enough to be performed either on the cloud, constrained devices or Machine-to-Machine (M2M) gateways. M2M gateways means that processing is done automatically, without requiring human intervention.

- Interpreting sensor data and infer new knowledge by reusing domain knowledge expertise. Reusing domain knowledge (e.g., ontology) is highly recommended [Simperl, 2009] [Suárez-Figueroa, 2010] [Suárez-Figueroa, 2010]. The interoperability of domain knowledge enables building cross-domain expertise.

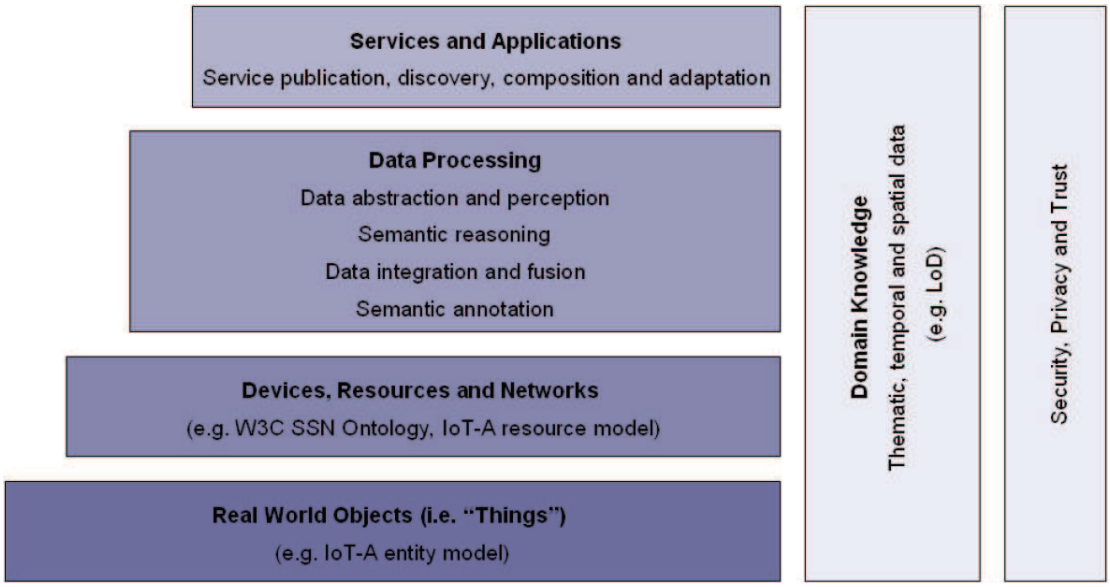

Figure 1.3: Semantics is required in the IoT according to [Barnaghi et al., 2012b]

- Securing IoT applications when designing these applications. This challenge should be solved using the same approach as for the two previous challenges by combining security knowledge expertise to find the most suitable security mechanisms to secure IoT applications.

1.4 Assumptions

In this thesis, we assume several facts. Firstly, we assume that a pre-treatment is done to clean sensor data (e.g., delete data unreliable in case a sensor is dying) and a pre-precessing is done to add sensor metadata such as measurement type, unit, sensor type, value and domain. For instance, sensor data is represented in this way: the domain is health, the measurement type is temperature, the value is 39 and the unit is degree Celsius. Secondly, we consider that sensor data is generated by small and simple sensors such as thermometer and not by complicated sensor such as electrocardiogram (ECG). Finally, we are not considering the following challenges: real-time and scalability/big data analytics.

1.5 Research Hypotheses

In this section, we provide research hypotheses that we address in this thesis:

- Hypothesis 1: The semantic engine is not too resource consuming. This is a first step to later embed the semantic engine in constrained devices. We are expecting that it

takes few minutes to run the semantic engine. This is done by measuring software performances which is explained in Section 3.7.

- Hypothesis 2: The semantic engine is generic enough to support various kind of IoT measurement. We are interested in checking that the semantic engine is adapted to heterogeneous IoT domains such as healthcare, smart home, smart cities, weather forecasting, smart car, etc. and handles the data generated by simple IoT devices. This is done by running M3, more precisely, S-LOR on datasets with different kind of measurements which is explained in Section 3.7.
- Hypothesis 3: The semantic engine enables building cross-domain IoT applications. This is done by running M3, more precisely, S-LOR on datasets with different kind of measurements and by loading cross-domain templates which is explained in Section 3.7.
- Hypothesis 4: Users are interested to integrate semantic web technologies to Internet of Things. This is done by looking at the visitor map and Google Analytics on our proof of concept available on the Web which is explained in Section 3.7. We keep it as an hypothesis even if people do not consider it as a research hypothesis. For us, it is important to evaluate whether it encourages persevering in this research topic.
- Hypothesis 5: The dataset of M3 rules is reliable to interpret IoT data. This aspect is important to ensure the reliability of the results provided by the reasoning engine. This is done by looking at completeness and correctness of the rules which is explained in Section 4.8.
- Hypothesis 6: A dataset of ontology-based projects relevant for IoT can be exploited outside of the M3 framework. It shows that the catalogue is relevant and exploited by people. This is done through a user form which is explained in Section 4.8.
- Hypothesis 7: The knowledge base built to interpret IoT data encourages the interoperability of data and domains. This can be done by following semantic web best practices which is explained in Section 4.8. It is important to encourage the reuse of our knowledge base.
- Hypothesis 8: The security knowledge base is built using the same methodology that for the M3 interoperable domain knowledge. It shows that M3 is generic enough for other domains such as security which is explained in Section 5.5.
- Hypothesis 9: A security knowledge base help non-experts in security choose security mechanisms fitting their needs to secure IoT applications. This is done through a user form which is explained in Section 5.5.

1.6 Contributions

To assist developers in designing and implementing cross-domain Semantic Web of Things (SWoT) applications, we devised the Machine-to-Machine Measurement (M3) framework.

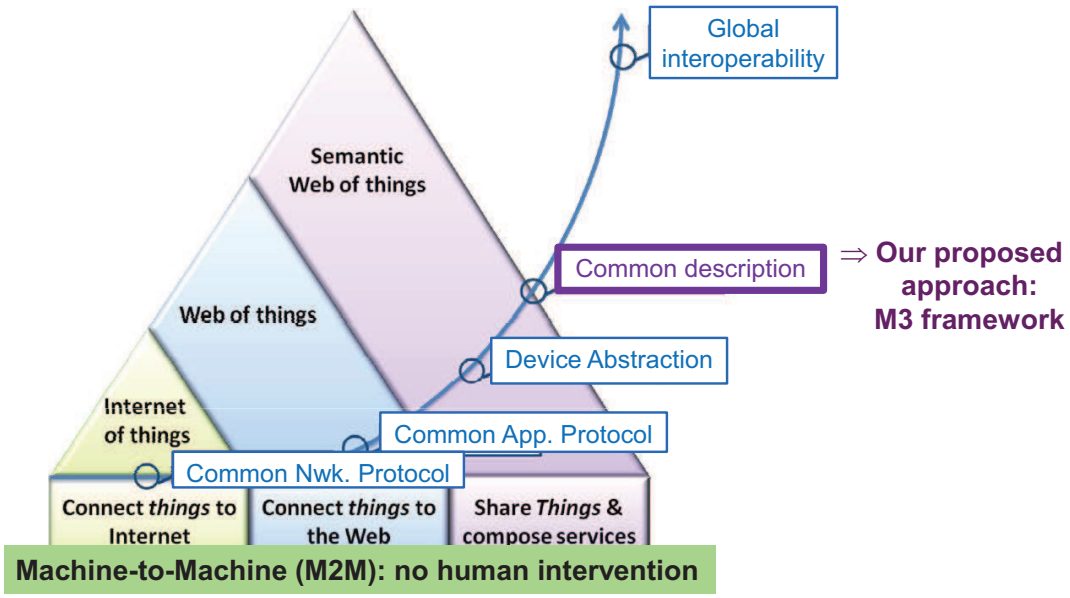

Figure 1.4: Next challenges of Semantic Web of Things [Jara et al., 2014]

Recently, Jara et al. explained that the next challenging tasks of Semantic Web of Things (see Figure 1.4) are: (1) a common description for sensor data, and (2) agreeing on a common catalogue of ontologies to annotate sensor data in an interoperable manner. Such challenging tasks have been resolved thanks to our proposed M3 framework [Jara et al., 2014].

The first contribution of this thesis is the **M3 framework which enables to automatically interpret sensor data and generate interoperable domain-specific or cross-domain SWoT applications**. In this thesis, we explain in details the components of the framework and how they are interconnected with each other. Moreover, the approach proposed has be integrated in a Machine-to-Machine (M2M) platform, called Com4Innov, deployed in Sophia Antipolis, France. A perspective was to extend this platform with new functionalities such as adding intelligence to data with semantics. For this reason, we demonstrate that this framework is compliant with standards such as ETSI M2M [M2M, 2012]. The distributed semantic-based M2M architecture that we propose is inspired by ETSI M2M standard. Several use cases such as naturopathy, tourism, transportation and security have been implemented as proof-of-concepts on the cloud or embedded on constrained devices such as mobile devices or tablets.

Our second contribution is an innovative concept for sharing, reusing and combining rules to interpret sensor data that we called **Sensor-based Linked Open Rules (S-LOR)**. Thanks to S-LOR, we provide a unified and interoperable mechanism to reason on sensor data. We semantically annotate SenML sensor data with semantics. We propose the M3 nomenclature implemented as an ontology to aggregate and describe in a unified way common sensors, measurements, units and domains. We interpret sensor data and enrich it to infer higher-level abstractions by exploiting the domain knowledge expertise.

This knowledge is extracted from the Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset that we have built. The goal of the LOV4IoT dataset is to reference, synthesize, classify and reuse more than 270 domain-specific projects relevant for IoT. Due to several interoperability issues to interlink the domain knowledge, we had to re-design the interoperable M3 domain knowledge composed of ontologies, datasets and rules. This task was essential to show the entire chain of the M3 processing. Indeed, it eases the reasoning on sensor data and facilitates the interlinking of cross-domain knowledge (e.g., weather and transportation, weather and smart home) to generate advanced and promising IoT applications.

Our third main contribution is the **Security Toolbox: Attacks and Countermeasures (STAC)** that assists users in finding the most relevant security mechanisms to secure IoT applications. The STAC approach reuses and adapts several components provided by the M3 framework. For instance, we have designed STAC using the same approach as for the M3 interoperable domain knowledge. STAC is a security knowledge base that classifies attacks and countermeasures in various domains such as sensor networks, wireless networks, network management, web applications, etc.

The three main contributions are validated through five use cases. The goal of the first use case is to show that the M3 framework can be used by developers and that is flexible enough to be embedded on Android-based constrained devices. In the second use case, we show how M3 can be used in the context of smart cars. In the third use case, we demonstrate how M3 can be embedded in smart fridges. In the fourth use case, we explain how M3 can be embedded in smart luggage. Finally, in the fifth use case we explain how STAC can help non-security expert developers to secure their applications.

1.7 Organization of Thesis

This thesis is organized as follows. Part I is composed of two chapters: 1) this introduction chapter in Chapter 1, and (2) the state of the art on Semantic Web of Things and the main challenges in Chapter 2. Part II is composed of three chapters focussing on our three main contributions: (1) the M3 framework to assist developers in designing IoT applications in Chapter 3, (2) S-LOR to interpret IoT data in Chapter 4, and (3) STAC a security knowledge base to assist developers in securing their applications in Chapter 5. Part III is composed of two chapters: (1) use cases employing our proposed approaches in Chapter 6, and (2) conclusion and future work in Chapter 7.

Chapter 2

State of the Art: Semantic Web of Things (SWoT) & Related Research fields

"The mind seeks and it is the heart that finds."

George Sand

"The more I learn, the more I learn how little I know."

Socrate

In this chapter, we investigate the Semantic Web of Things (SWoT) and related research fields. We start by giving basic definitions of Ubiquitous Computing (UbiComp), Pervasive Computing, Ambient Intelligence (AmI), Context-Awareness, Ambient Assisted Living (AAL), Smart Homes, Semantic Sensor Networks (SSN), Machine-to-Machine (M2M), Internet of Things (IoT), Web of Things (WoT), Semantic Web of Things (SWoT), Smart Cities and Physical-Cyber-Social Computing (PCS) in section 2.1. We then highlight some of the main challenges to overcome in section 2.2. Afterwards, we scrutinize related works and tools for each challenge in section 2.3: (1) interoperable IoT data, (2) interpreting IoT data, (3) inter-domain interoperability, (4) assisting IoT developer tasks, (5) 'Sensor Plug & Play' mechanisms, (6) semantics applied to constrained devices, and (7) security concerns for IoT. Finally, we conclude the chapter by summing up the current limitations of existing works and elucidate some solutions to overcome these limitations in section 2.4.

We synthesize in Figure 2.1, the main projects involved in Semantic Web of Things or Semantic Sensor Networks, their relationships with each other and the tools they designed. We analyze most of these projects and tools below.

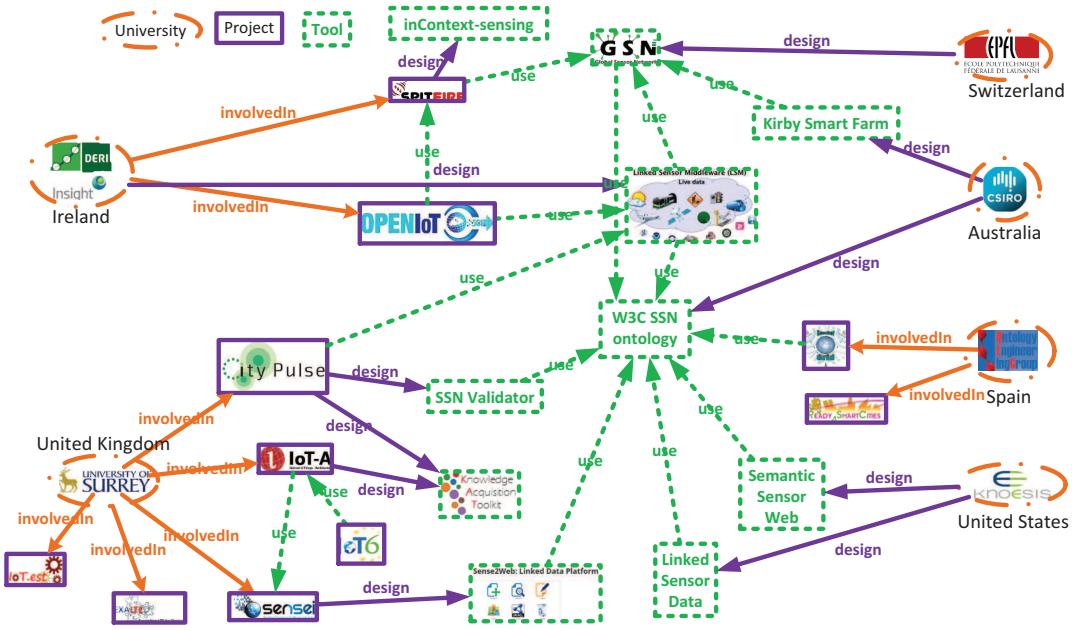

Figure 2.1: Semantic Web of Things projects overview

2.1 Understanding Semantic Web of Things Related Research Fields

In this section, we explain the main challenges of heterogenous research fields having overlapping goals: Ubiquitous Computing (UbiComp), Pervasive Computing, Ambient Intelligence (AmI), Context-Awareness, Ambient Assisted Living (AAL), Smart Homes, Semantic Sensor Networks (SSN), Machine-to-Machine (M2M), Internet of Things (IoT), Web of Things (WoT), Semantic Web of Things (SWoT), Smart Cities and Physical-Cyber-Social Computing (PCS). The evolution of Ubiquitous Computing has given place to new terms like 'Pervasive Computing', 'Context-aware Computing', 'Mobile Computing', 'Wearable computing' and now 'Internet of Things'.

2.1.1 Ubiquitous Computing (UbiComp)

Ubiquitous computing is a research field aiming at integrating computers into objects.

In 1993, Weiser introduces the notion of ubiquitous computing [Weiser, 1993].

In 2003, Chen et al. integrate semantic web technologies to pervasive computing [Chen, 2004]. They design the **Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA)** ontology to describe user profiles, beliefs, desires, etc. [Chen et al., 2003] [Chen et al., 2004] [Chen et al., 2005b] The SOUPA ontology is integrated in the **Context Broker Architecture (COBRA)** architecture to build smart meeting rooms [Chen, 2003]

[Chen et al., 2005a] [Chen, 2004]. COBRA is a centralized architecture for context-aware systems in smart environment. Then, the authors developed EasyMeeting, an intelligent meeting room based on the COBRA architecture. They define a policy language for users to control the sharing of their information and two ontologies SOUPA and COBRA-ONT. The ontology COBRA-ONT is for modeling context in an intelligent meeting room: places, agents, agents location and agent's activity.

The **CONON** ontology [Wang et al., 2004] [Gu et al., 2004] has been designed and integrated to the **Service-Oriented Context-Aware Middleware (SOCAM)** [Zhang et al., 2005] [Wang et al., 2002] architecture. Neither prototype nor ontologies are available online. The SOCAM architecture is an OSGibased architecture that converts various physical spaces where contexts are acquired into a semantic environment where context-aware applications can share and access them easily. The context ontology CONON [Wang et al., 2004] [Gu et al., 2004] defines several concepts like computational entities (services, applications, devices), location, person, activity and indoor space (building, room, corridor and entry).

In 2006, Hilera et al. survey the ontologies in ubiquitous computing: SOUPA, CONON, FIPA and GUMO [Hilera and Ruiz, 2006].

Jeong et al. [Jeong et al., 2006] propose in the **Ubiquitous Computing Architecture (UTOPIA)** project, an ontology to describe: (1) environment (e.g., humidity, temperature), (2) objects with person, computing device (computer, display, printer), environmental devices (e.g., air conditioner, curtain, door, window), (3) space represents building or room, (4) activities, (5) preferences, and (6) people. Within the UTOPIA project, some services such as U-Restaurant Service (some information are provided about religious beliefs, if the meal is vegetarian), U-Museum service, and U-Theme Park service have been developed.

2.1.2 Pervasive Computing

Pervasive computing integrates computers into the life of everyday users to understand the surrounding environment. Their work on the dynamicity and communication between devices.

In 2003, Ranganathan et al. design ontologies for the **GAIA** framework [Ranganathan et al., 2003]. Their ontologies are written with the DAML+OIL language. DAML+OIL has been superseded by Web Ontology Language (OWL). OWL is a W3C recommendation since 2004 [Welty et al., 2004].

In 2006, Henricksen et al. want to design applications that are flexible, adaptable, and capable of acting autonomously on behalf of users [Henricksen and Indulska, 2006]. In their paper, they support the software engineering process of context-aware pervasive computing applications to facilitate the development of flexible and evolvable software.

In 2006, Baumgartner et al. survey ontologies for situation awareness: SOUPA, CONON, SAWA and Situation ontology in [Baumgartner and Retschitzegger, 2006]

In 2007, Bibakis et al. survey 19 pervasive computing systems [Bikakis et al., 2008]. They compare existing works according to the modeling (OWL, RDF, DL, DAML+OIL), reasoning (RDQL, DL, Jess, Bayesian), architecture (centralized or decentralized) and context-aware services proposed.

In 2007, Ejigu et al. integrate rule-based reasoning in pervasive computing [Ejigu et al., 2007].

In [Chaari et al., 2007], they highlight the lack of standard and reusable reference model that can be used to handle context in multiple domains of applications. They use SWRL rules and the Jena reasoning engine for their decision rules. They use the Petri network to adapt the application to new contexts.

In 2007, Coyle et al. merge sensor data and semantically annotate it with RDF via XSLT [Coyle et al., 2007]. They conceived a top-level ontology to facilitate the capture of domain knowledge [Ye et al., 2011] and the Ontonym ontology for pervasive systems [Stevenson et al., 2009]. In 2015, Ye et al. survey semantic web technologies in pervasive computing where they compare reasoning mechanisms: RDF, OWL Full, OWL DL, OWL 2 DL, OWL 2 EL, OWL 2QL OWL 2RL, SPARQL, SWRL and DL-Safe rules [Ye et al., 2015]. They conclude their survey by explaining the benefit of semantic web technologies. These technologies facilitate the unambiguous sharing and understanding of domain knowledge across heterogeneous and distributed systems. They introduce the challenge of commonly-agreed ontologies that would enable the standardize description of our surrounding environment and encourage combining ontologies with rules.

2.1.3 Ambient Intelligence (AmI)

Ambient Intelligence (AmI) focuses on Activity Recognition and Human Computer Interaction (HCI).

In 2004, Preuveneers et al. design an ontology to define a context terminology to be understood by all participating devices in ambient intelligence [Preuveneers et al., 2004]. The ontology defined sensors and the related rules such as turn on/off the lights according to the weather (cloudy, rainy) or if the person is located in the room. This work is integrated in the **Context-Driven Adaptation of Mobile Services (CODAMOS)** project.

In 2010, Antoniou et al. describe a reasoning framework for ambient intelligence that integrates rule-based reasoning on top of ontology-based context models [Patkos et al., 2010]. In 2012, Vasileios et al. work on real-time activity recognition [Vasileios and Antoniou, 2012] for Ambient Intelligence. Regarding the activity recognition part, their approach is based on machine learning to deal with real-time and incomplete data or contradictory information. They integrate case-based reasoning based on the k-Nearest Neighbors (kNN) algorithm. However, they describe activities in ontologies and the semantics is expressed in the machine learning dataset. They applied their approach to two use cases: smart classroom and Ambient Assisted Living where they they design SWRL rules such as high noise level, high temperature level, high lighting level, low lighting level, projector is on, bathroom door closed, lie in bed, TV on, etc. They design the assisted living ontology and define relationships between light, activities and season. Such ontologies could be re-used in other contexts such as tourism.

In 2012, **Smart Building Ontology for Ambient Intelligence (BOnSAI)** is a new ontology for ambient ontology [Stavropoulos et al., 2012].

In 2014, Rodriguez at al. survey ambient intelligence and context awareness [Rodríguez et al., 2014]. In their survey, they are focused on knowledge-based techniques, more precisely, ontologies to track human activities.

2.1.4 Context-Awareness

Context-Awareness focuses on understanding and adapting the surrounding environment. It takes into account spatial and temporal aspects in the reasoning mechanism to adapt the behavior of context-aware applications.

In 2004, Christopoulou et al. design the **GAS ontology** to compose ubiquitous computing/context-aware applications [Christopoulou et al., 2004] [Christopoulou and Kameas, 2005].

In 2010, Riboni et al. survey the reasoning techniques applied to context-awareness [Bettini et al., 2010]. After that, they design **COSAR**, an activity recognition system where the ActivO ontology combined to statistical classification is used to deduce activities [Riboni and Bettini, 2011]. Their innovative hybrid reasoning algorithm is running on the mobile device. They can deduce 10 activities: brushing teeth, hiking up, hiking down, riding bicycle, jogging, standing still, strolling, walking downstairs, walking upstairs and writing on black board.

In 2011, Barbero et al. apply context aware reasoning to Internet of Things [Barbero et al., 2011]. New challenges are to: (1) deal with heterogeneous data, (2) be scalable to large scenarios and to resource constrained devices, and (3) manage interoperability.

In 2012, Chahuara et al. interpret sensor data though SWRL reasoning in the building home automation domain to improve comfort and autonomy at home [Chahuara et al., 2012].

Sorrentino et al. design the Context Aware Infrastructure (CAFE), a contextual infrastructure that exploits the integration of Semantic Networks and the Object-Oriented model [Sorrentino, 2012]. They deal with incompleteness of data from sensors. The inference and reasoning is done through SPARQL. SPARQL is a language to query data, the SPARQL engine is not really a reasoning engine. Usually, reasoning engines add new triples to the knowledge base. As a future work, they claim that it would be beneficial to integrate an ontology to their model.

In 2012, Wongpatikaseree use ontologies in context-awareness to recognize current human activities [Wongpatikaseree et al., 2012]. They develop the **Ontology Based Activity Recognition (OBAR)** system to recognize 13 activities: 'Sitting & Relaxing', 'Sitting on the toilet', 'watching TV', 'working on computer', 'eating or drinking', 'sleeping', 'lying down & relaxing', 'wash dish', 'take a bath', 'cooking', 'make a drink', 'sweep the floor' and 'scrub the floor'. Theses rules are based on Description Logic (DL), implemented with the Semantic Web Rule Language (SWRL) and defined in the ontology as `owl:Restriction`.

Perera et al. survey context-awareness with a focus on Internet of Things [Perera et al., 2014]. They highlight the necessity to interpret, analyze and understand sensor data to perform machine-to-machine communications in IoT. They classify six techniques such as supervised learning, unsupervised learning, rules, fuzzy logic, ontological reasoning and probabilistic reasoning. Further, they clearly explain pros and cons and sum up them in a table. According to their table, rule and ontology-based techniques contain few cons. The main shortcomings of these two-techniques is that the rules must be manually created, which can be error-prone and that there is no validation or quality checking. Stemming from the 'Linked Open Data' approach, there is a need to have a dataset of rules which can be shared, reused and validated by domain experts. With such approach, rules are only defined once in an interoperable manner. Pros regarding rule-based systems is that rules are simple to define, easy to extend and require less computational resources.

2.1.5 Ambient Assisted Living (AAL)

Ambient Assisted Living (AAL) is based on Ambient Intelligence. AAL gives the opportunity to elderly to stay independent. Semantics will enable through reasoning to assist old persons with their daily routine by understanding their activities.

In 2009, Chen et al. design ontologies for activity recognition [Chen and Nugent, 2009] [Chen et al., 2013]. They recognize 7 activities: make coffee, have bath, watch TV, make chocolate, brush teeth, make tea and make pasta. In the **MobileSage** project, Skillen et al. use semantic web technologies for user modelling and personalisation reasoning. The reasoning is a rule-based mechanism based on the SWRL rule language [Skillen et al., 2013]. Compared to previous approaches, their work on composite activity recognition such as interleaved and concurrent activities. Ontological reasoning is used for simple activity recognition and they include temporal inference to support composite activity recognition [Okeyo et al., 2013].

In 2012, Zografistou et al. design four ontologies: person, core, health assisted living [Zografistou, 2012]. They integrate two kind of reasoning in their system: rule-based reasoning and case-based reasoning. Rule-based reasoning is used to take into account person's health and the case-based reasoning to deduce activities. The rule-based reasoning is based on the SWRL language and the Jess implementation engine.

2.1.6 Smart Homes

Smart homes and domotic are mainly focused on reducing energy consumption and improving the comfort of inhabitants.

In 2008, Bonino et al. design the **DogOnt** ontology for domotic environments [Bonino and Corno, 20
Their rule-based mechanism is still based on the Semantic Web Rule Language (SWRL).
They take into account the description of actuator states (e.g., on and off). In [Bonino and Corno, 2010],
the authors are focused on the reasoning part; they use Jena rules to implement SWRL rules.
Bonino et al. use ontologies to estimate power consumption is smart homes [Bonino et al., 2014a].
Bonino et al. design the Domotic OSGi Gateway (Dog) Gateway, an ontology-powered middleware based on the OSGi framework and the DogOnt ontology to support the integration of different networks and support logic-based intelligence [Bonino et al., 2014b]. Dog is an open-source solution capable of running on low cost hardware such as Rasperry Pi. The dog gateway's architecture is composed of four layers: (1) Drivers layer that provides an interface to the various home and building automation networks to which dog can be connected and handles network-specific protocols, (2) Core layer that contains the core intelligence based on the dog ontology, (3) Addons layer that contains the data storage, stream processing, rule-engine, and (4) Communication layer that offers access to external applications either by rest endpoint or web sockets This work is focused on smart buildings and has not be applied to other domains.

In Austria, Kofler et al. propose the **ThinkHome** ontology [Reinisch et al., 2011] [Kofler et al., 2012], where they classify: (1) nonrenewable or renewable energy, (2) energy providers, (3) energy tariffs, (4) energy facilities, and (5) energies properties [Kofler et al., 2011]. Their prototype propose a self-regulation of heating and cooling system tailored to schedule: night-time, weekends, holidays, seasons. Then, this work has been improved with Staroch

et al. who design the **SmartHomeWeather** ontology for smart homes and related to the weather [Staroch, 2013]. This ontology deduces if there is a need to irrigate the garden, when we can open the windows and when do we have to keep them shut or even when do we need sun protection? Such reasoning is possible since they define numerous concepts related to weather sensors such as temperature, humidity, dew point, wind speed and direction, precipitation intensity and probability, atmospheric pressure, cloud cover, solar radiation, suns position. Further, in the ontology, numerous SWRL rules are designed to interpret sensor measurements as `owl:Restriction` and could be re-exploited to be combined with other works related to weather. They follow the 'Methondology' methodology to design the ontology.

TNO designs the **Smart Appliances REFerence (SAREF)** ontology, an unified ontology for smart appliances in the smart home domain [Appliances, 2013]. They cover popular sensor and actuators. To the best of our knowledge, we did not find any methods to interpret sensor data based on this ontology. This is a new ontology in the smart home among the 45 ontologies for smart homes that we referenced.

2.1.7 Semantic Sensor Networks (SSN)

Semantic Sensor Networks (SSN) is focused on the interoperability of the description of physical sensor networks to later ease the sensor discovery. Further, they work on semantically annotating sensor data to release it as 'Linked Sensor Data'.

Sheth et al. design **'Semantic Sensor Web'** in 2008 to semantically annotate sensor data with the Semantic Web languages such as Resource Description Framework in Attributes (RDFa) [Sheth et al., 2008]. They define an ontology (sensor-observation.owl) to support interoperability over heterogeneous environments, and a second ontology (weather.owl) to describe concepts and units related to the weather. Further, they introduce the idea to reason over semantic sensor data to infer high-level abstraction in two domain-specific scenarios: weather or healthcare. For instance, in the weather domain, they can deduce PotentiallyIcy, Freezing, Blizzard, LowVisibility and HighWinds.

Both [Compton et al., 2009] and [Bell et al., 2009] discussed the state of the art for the semantic specification of sensors. There are a great deal of sensor ontologies, few of them are linked with each other and are available online.

The **W3C Semantic Sensor Networks (SSN) ontology** is a synthesis of all of these sensor ontologies. The authors review all of these sensor ontologies[1] and compare them according to different criteria: (1) purpose of the ontology, (2) status of the ontology: online, documentation, maintained, etc., (3) key concepts found in the ontology, (4) adoption of the ontology, (5) level of sophistication, and (6) weakest features.

The SSN ontology is based on the following works:

- The CESN ontology [Calder et al., 2010]
- The OntoSensor ontology, a prototype sensor knowledge repository [Russomanno et al., 2005a] [Russomanno et al., 2005b] [Goodwin and Russomanno, 2006]

[1] `http://www.w3.org/2005/Incubator/ssn/wiki/Review_of_Sensor_and_Observations_Ontologies`

- A service-oriented ontology for Wireless Sensor Networks (WSN) [Kim et al., 2008]
- An ontology for sensor networks describing the topology, network setting, sensor properties, dataflow and sensor network performance [Jurdak et al., 2004].
- Other ontologies related to sensor networks [Eid et al., 2007] [Eid et al., 2006] [Avancha et al., 200

These ontologies are mainly focused on the description of physical sensor networks such as sensor capabilities, location of the sensor (latitude and longitude), etc. They do not address the problem to describe sensor data in an interoperable manner to ease the reasoning task.

SSN defines high-level concepts for representing sensors, their observation and the surrounding environment. However, according to the W3C SSN ontology final report[2], SSN has several limitations:

- It does not provide common terms to represent subclasses of `ssn:Sensor` type, measurement type, unit type or domain type (`ssn:FeatureOfInterest`). This shortcoming hinders machine automation.
- According to the ontology documentation[3], "the W3C SSN ontology does not describe domain concepts, time, locations, etc. These concepts are intended to be included from other ontologies via OWL imports. This leads to interoperability issues between domain ontologies, since domain ontologies relevant for IoT are not standardized.
- Future work is to "standardize the SSN ontology to use it in a Linked Sensor Data context" and to "standardize the SSN ontology to bridge the Internet of Things". These facts highlight the need to provide a common description of sensor measurements. Adapting the SSN ontology to Internet of Things is one of the purposes of this thesis.

The SSN ontology leverages the **Sensor Web Enablement (SWE)** [Botts et al., 2008] [Bröring et al., 2011a] standard which provides high-energy consuming services such as being alerted when a specific event occurs or asking for more detailed measurements. SWE is difficult to set up, configure and deploy. The information layer defined by the SWE is composed of :

- SensorML for describing sensors (e.g., a sensor on a bottle of milk)
- O&M is an encoding for data observed or measured by sensors (e.g., the bottle of milk has an expiration date).
- SOS (Sensor Observation Service) which provides access for discovering and retrieving sensor data (e.g., the bottle of milk).
- SAS (Sensor Alert Service) which is used for alerting clients once an event they are interested in occurs (e.g., the expiry date of the bottle of milk is tomorrow).

[2] `http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/#Directions_for_future_work`
[3] `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#`

- SPS (Sensor Planning Service) is for being informed that an event occurred via SMS (Short Message Service), email or phone.
- WNS (Web Notification Service) is for managing sensors remotely (e.g., switch off the fridge).

2.1.8 Machine-to-Machine (M2M)

Machine-to-Machine (M2M) is a subset of IoT enabling machines or devices to communicate with each other without human intervention. The main challenge of M2M is to provide interoperability among heterogeneous communication protocols.

The ETSI standardization advocates the **ETSI M2M architecture** [Boswarthick et al., 2012], as depicted in the Figure 2.2:

- M2M area networks are composed of: (1) M2M devices such as sensors (e.g., a thermometer), RFID tags or EPC embedded on products such as foods, CDs, DVDs, actuators (e.g., to switch on/off the lamp), transducers (e.g., keyboard) and controllers (e.g., accelerometer), and (2) M2M gateways which aggregate heterogeneous data from M2M devices.
- M2M applications are accessible through RESTful web services. M2M applications are deployed in heterogeneous domains such as home networking [Zhang et al., 2011], healthcare [Jung et al., 2012] and vehicular networking [Booysen et al., 2012].
- M2M service capabilities enable users through devices (PC, laptop, mobile phone, PDA) accessing the M2M applications.

ETSI M2M explains that interpreting and using M2M data from heterogeneous sources is considered essential for creating high-level M2M applications but do not provide any concrete approaches for this task [M2M, 2012]. Interoperability across different M2M domains and staying independent from vertical markets are essential to build innovative applications. They propose the idea to re-use sensor data across different applications. Securing the ETSI M2M architecture is a difficult task, since there is a need to secure heterogeneous wireless communications (cellular, wireless, wired), devices (sensor or mobile phone) and applications (programming language, framework, database).

Swetina et al. explain that industries use proprietary systems that make it difficult to extend applications, integrate new data and interoperate with other solutions [Swetina et al., 2014]. One of the goals of oneM2M, the international standard for M2M, is to: (1) shorten time to market by removing the need to develop common components, and (2) simply development of applications by providing a common set of Application Programming Interface (APIs). The oneM2M architecture comprises of: (1) common services entity (CSE), (2) application entity (AE) that provides application for end-users such as blood sugar monitoring, and (3) common services functions (CSFs) where components can be used in other CSE and applications. For instance, CSF components can be: (1) data management and repository (DMR) that could handle the processing of M2M data, (2) security (SEC) to enable secure establishment of service connections and data privacy. **oneM2M** is a necessary standard

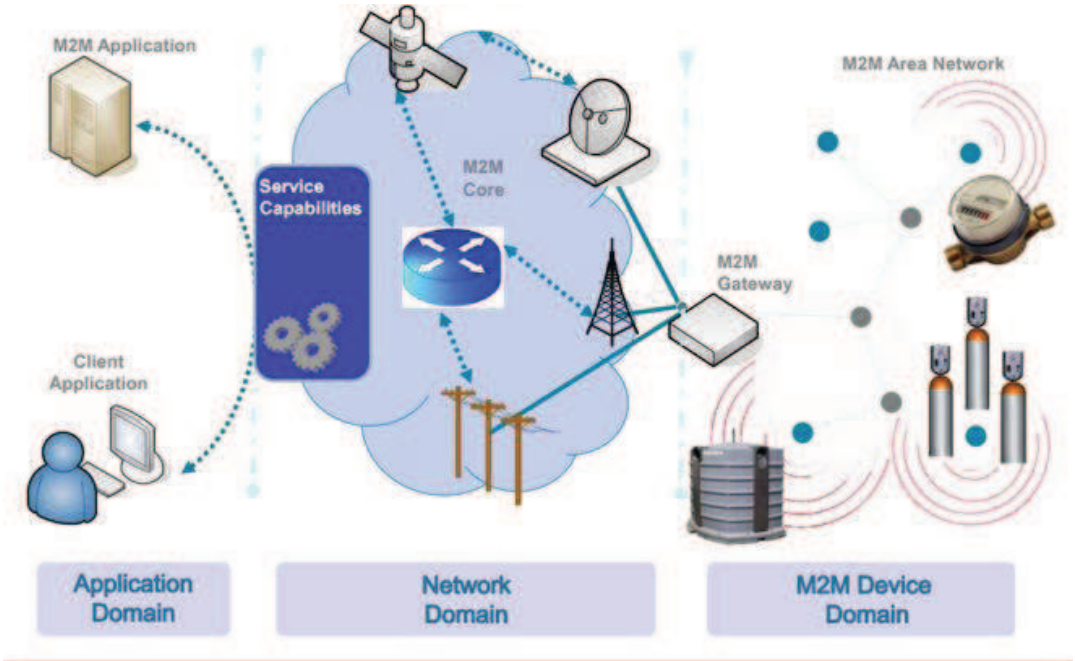

Figure 2.2: ETSI M2M architecture [Boswarthick et al., 2012]

to ensure data interoperability, and efficient development of M2M systems in various domains [OneM2M and Requirements, 2014]. The working group 5 (WG5) of oneM2M focus on Abstraction and Semantics [OneM2M et al., 2014]. oneM2M, explains the need to secure sensitive IoT data regarding the: (1) storage, (2) communications and (3) applications [OneM2M and Security, 2014].

2.1.9 Internet of Things (IoT)

At the beginning, the vision of **Internet of Things (IoT)** was to attach RFID tags on objects. Then, the novelty of Internet of Things (IoT) is to connect objects or things to Internet, hence connecting the physical world to the virtual world. Recent challenges are: (1) to connect heterogeneous domain with each other 'to break vertical silos', and (2) interpret IoT data. IoT is also a new fashionable way to talk about M2M.

Aztori et al. survey Internet of Things [Atzori et al., 2010]. They describe main security research challenges for IoT [Atzori et al., 2010].

Miorandi et al. clearly explain a lack of standardization related to models and data formats and describe the need for: (1) cross-domain applications, (2) semantic interoperability and data management for exchanging and analyzing IoT/M2M data to infer useful information and to ensure interoperability among IoT applications and for reasoning, and (3) security and privacy by design for IoT architecture, more precisely, data confidentiality

and trust (e.g., using RBAC for access rights) [Miorandi et al., 2012].

Barnaghi et al. define four interoperability issues [Barnaghi et al., 2013]:

- *Technical interoperability* that concerns heterogeneous software and hardware.
- *Syntactical interoperability* that concerns data formats. This is important to later interpret IoT data and build smartness applications. They underline the need to agree on common vocabularies to describe data.
- *Semantic interoperability* that concerns interpretation of meaning of data exchanged.
- *Organizational interoperability* that concerns heterogeneity of the different infrastructures.

Further, they explicitly indicate the need to assist users by designing tools to build IoT applications. Finally, they addressed security and privacy issues related to IoT data. In this thesis, we address the syntactical and semantic interoperability challenges.

Lee et al. explain that one of the fundamental characteristics of the IoT will be the management of the data produced and its interpretation for application purposes [Lee et al., 2013]. Gubbi et al. explicitly describe "novel fusion algorithms need to be developed to make sense of the data collected" [Gubbi et al., 2013].

In 2014, Chen et al. introduce the need for intelligent processing for IoT data and explain the issue related to **domain specific-applications**: applications cannot combine the data from different silos ("cannot correlate and integrate the data from different silos and getting the data from heterogeneous sources") [Chen et al., 2014]. They explain that IoT applications will be deployed in nine domains such as smart home, smart medical care, smart agriculture, intelligent transportation, smart environmental protection such as air or water quality monitoring, smart safety such as food safety monitoring, smart grid such as smart power, domain industry application such as chain tracking and smart logistics such as inventory control or traceability.

According to Qin et al., one of the current major challenges in IoT is to connect real world object to the Web, to collect their data and process them to build intelligent applications [Qin et al., 2014]. This challenge will be achieved by things themselves and not humans. To resolve this challenge, they explain that semantic enrichment of sensing data is a promising research direction. Another important aspect mentioned is the need to extract useful knowledge from the data, called 'Knowledge Discovery'.

Xu et al. explain that sensor data are useless if we do not analyze and understand them, which is a challenging task for end users since they do not have the skills required [Xu et al., 2014]. Further, they introduce the challenging task to **combine heterogeneous sensor data with traditional data to build practical applications** for industries. They do not provide any solutions for this part.

Desai et al. explain the interoperability issues between **vertical applications** and the need of standardization of the data models [Desai et al., 2014]. In [Desai et al., 2014], they design the Semantic Gateway as Service (SGS), a bridge between sensors and end-user applications. They integrate the W3C SSN ontology, the SemSOS tool and domain ontologies in their gateway to semantically annotate sensor data. They explicitly describe that for domain specific applications, the gateway can be equipped with additional domain specific

ontologies. There is a need of an interoperable cross-domain knowledge to fulfill these limitations. However, they explain that the W3C Semantic Sensor Networks (SSN) ontology is sufficient to provide a basis for high level abstraction. We are not agree with this fact, sensor measurements can be describe as literals, but they should be describe as URIs when they referred to the same type (e.g., precipitation) to avoid any ambiguities. They explain that they annotate raw sensor data using SSN and domain ontologies, but do not clearly explain which ones. They do not underline the difficulties to reuse domain ontologies relevant for IoT and interoperability issues. Further, they address security issues, by integrating OAuth authentication server to ensure the authorization security property which allows the users to control sensor data in the gateway. The gateway supports heterogeneous lightweight application protocols such as CoAP, XMPP and MQTT. A Raspberry Pi or Arduino can be considered as a gateway. Services such as Xively enable providing graphical user interface on such gateways.

In 2015, Ganz et al. clearly explained that interpreting data in a meaningful way and how actionable information can be extracted from the raw IoT data is still an open challenge [Ganz et al., 2015].

Internet of Things Environment for Service Creation and Testing (IoT.est)[4] explains that existing applications are domain-specific and highlights the lack of interoperability between the different silos since they use heterogeneous communications, technologies, protocols and formats. They also introduce the necessity of M2M-oriented solutions to secure IoT architectures and applications.

IERC (European Research Cluster on Internet of Things)[5] develops a global interoperability framework. It explains that IoT architectures are confined to particular domains.

Ozpinar explains that resolving the meaning of data is a challenging problem and without processing it, the data is useless [Öspinar, 2014]. He also discusses the challenging problems regarding heterogeneity of billions of devices and he outlines that the challenge of resource-constrained devices has to be taken into account.

2.1.10 Web of Things (WoT)

Web of Things (WoT), an evolution of IoT, provides interoperability between hardware devices and communication or application protocols but does not add intelligence to things yet. The novelty is to access to devices and produced data through the Web. Existing WoT platforms deal with heterogeneous protocols and easily share sensor data on the Web. However, they do not use semantic web technologies.

W3C Web of Things[6] underlines the need to use semantics to: (1) ensure interoperability, e.g., as a basis for describing physical units, (2) encourage use of common vocabularies, (3) how should be standardized, and (4) interpreting sensor input.

In 2010, Guinard et al. introduce the new idea of **Web of Things (WoT)** to connect physical things to the web using the RESTful architecture [Guinard et al., 2010].

[4] `https://www.ecs.hs-osnabrueck.de/34734.html`
[5] `http://internet-of-things-research.eu/`
[6] `http://www.w3.org/2014/02/wot/`

Zeng et al. survey the Web of Things topic and highlight open issues such as: (1) heterogeneity of devices and communications protocols, (2) security and privacy, (3) sensor discovery, (4) context-awareness to adapt the environment according to the user profile, (5) Semantic Web Services adapted to the Web of Things [Zeng et al., 2011]. They conclude their survey by indicating that the smart things should speak the same language to communicate with each other.

2.1.11 Semantic Web of Things (SWoT)

Semantic Web of Things (SWoT) is a recent research field aiming to integrate Semantic Web technologies to Internet of Things. It is also an evolution of the Web of Things by integrating semantics. SWoT focuses on providing interoperability among ontologies and data.

In 2011, the **Spitfire** project introduced the idea to combine semantic web technologies and Internet of Things to build the **'Semantic Web of Things'** [Pfisterer et al., 2011].

Ruta et al. design the **Semantic Web of Things framework** which aims to enrich real-world objects with semantic annotations [Ruta and Scioscia, 2014].

Barnaghi et al. highlight the need of semantics to: (1) provide unambiguous IoT data descriptions to be interpreted by machines, (2) combine data from different physical worlds, (3) semantic reasoning, and (4) sensor discovery [Barnaghi et al., 2012b].

2.1.12 Smart Cities

Smart cities or environments is a new popular term. More and more projects also integrate semantics in this topic. They focus on real-time, 'Big Data' and scalability issues.

READY4SmartCities[7] is a project to reduce energy consumption and CO2 emission in smart cities by using ontologies and linked data [Raul Garcia-Castro, 2014]. This project provides guidelines to help data providers to generate energy-related data as Linked Data. It introduces the concept of cross-domain data such as climatic, occupation, pollution, traffic, activity, etc. It builds a dataset with 50 domain ontologies specific to smart cities and smart home. This project does not cover main IoT domains such as healthcare, smart farm, etc. Further, they do not encourage domain experts to share online and improve their ontology according to the best practices. Euzenat et al. underline limitations of current ontology mapping tools to automatically combine the domain knowledge relevant for the READY4SmartCities project [Euzenat, 2014]. They explain the need to share alignments, the results provided by ontology matching tools, so they can be reused and improved.

The **STAR-CITY** project is deployed in four cities: Dublin, Bologna, Miami and Rio [Lécué et al., 2014c]. They use semantic web technologies to diagnose and predict road traffic congestions [Lécué et al., 2012] [Lécué et al., 2014b] [Lécué et al., 2014a]. For their processing they use 6 heterogeneous sources: (1) road weather conditions, (2) weather information, (3) Dublin bus stream, (4) social media feeds, (5) road works and maintenance, and (6) city events. They use SWRL rules to define rules such as heavy traffic flow [Lécué et al., 2014b].

[7] `http://www.ready4smartcities.eu/`

The **CityPulse** project defines 101 scenarios such as public parking space availability prediction, real time travel planner, air pollution countermeasures, what is my route and efficient public transport [Barnaghi et al., 2014]. These scenarios underline the need of tools to design interoperable IoT applications. The CityPulse project is focused on large-scale analysis and real-time intelligence, two aspects that we do not handle in this thesis, but need to be considered as future work.

The **SmartSantander** project clearly explains that current deployments are closed and vertically-integrated solutions tailored to specific application domains [Sanchez et al., 2014]. In this project, they deploy 20,000 sensors measuring temperature, humidity, particle, Co, NO2 or monitoring parks and gardens irrigation, outdoor parking area management and traffic intensity monitoring [Gluhak et al., 2011]. These sensors have been deployed in five cities: Melbourne, Lubeck, Guildford, Belgrade and Arhus [Sanchez et al., 2011].

2.1.13 Physical-Cyber-Social Computing (PCS)

Physical-Cyber-Social Computing (PCS) focuses on interconnecting data generated by sensors (physical world) to the data from the Web (cyber world) and to the data from social networks (social world). In this thesis, we are considering the sensor data from the physical world and the background knowledge from the cyber world, but we are not considering the social-related data.

Sheth et al. mentioned the need to use the background knowledge found throughout the Web, but do not clearly explained that it is not so easy to combine and reuse heterogeneous domain knowledge since they are not structured in the same way [Sheth et al., 2013].

Ganz et al. explain the need of: (1) new (semi-)automatic techniques to interpret 'Cyber-Physical Data' and infer new knowledge, (2) taking into account context information, (3) a standardized model for meta-information, and (4) understand the meaning of data with both temporal and spatial attributes [Ganz, 2014].

2.1.14 Discussions

In all of these research fields, they design ontologies and try to understand the surrounding environment either with machine learning algorithms or rule-based reasoning. If their ontologies are shared online, we could reuse them, instead of reinvent the wheel each time. We have in mind to reuse and combine the ontologies and related reasoning that have been designed in ubiquitous computing, pervasive computing, ambient intelligence, context-awareness, Ambient Attested Living, Smart Cities and Smart Homes. Reusing and combining such ontologies and related reasoning should be easy since most of the time the standardized semantic web languages are used (RDF, RDFS and OWL). Further, a SWRL rule-based approach is frequently employed for the reasoning process in pervasive computing and related research fields. The presented approaches are not based on the open-source approach to share, reuse and combine the previous works. What is missing in all of these research fields is to have a universal reasoning which can be applied in all systems and though all domains. In the same time, we could help developers in designing interoperable applications to deploy in the cloud or even in smart devices. Further, it would enable

Research field	Acronym	Year	Features
Ubiquitous computing	UbiComp	1993	- Computing is everywhere and anywhere
Pervasive computing	-	2003	- Dynamicity - Communication between devices
Context-Awareness	-	2004	- Understand and adapt the environment (reasoning) - Spatial + temporal aspects
Ambient Intelligence	AmI	2004	- Activity recognition - Human Computer Interaction
Smart Homes	-	2008	- Reduce energy consumption - Improve human comfort
Semantic Sensor Networks	SSN	2008	- Interoperability to describe sensor networks - Sensor discovery - Semantically annotate sensor data ('Linked Sensor Data')
Ambient Assisted Living	AAL	2009	- Activity recognition - Help elderly people or people with deficiencies to stay independent - Social impact - Reduce hospital cost
Web of Things	WoT	2009	- Connect and interact with objects through the Web - Provide interoperability among application protocols
Machine-to-Machine	M2M	2011	- Interoperability among communication technologies
Internet of Things	IoT	-	- Attach RFID on objects (1999) - Provide Interoperability among domains (2012) - Interpret IoT data (2012)
Semantic Web of Things	SWoT	2011	- Integrate semantic web technologies in IoT - Interoperability among ontologies - Interoperability among data
Smart Cities	-	2012	- Scalability - Real-time
Physical-Cyber-Social	PCS	2013	- Interconnecting data

Table 2.1: Features of SWoT-related research fields

to the developers to avoid to rewrite the ontology and rules to interpret the surrounding environment, which is a time-consuming process. It would be interesting to see how to build a system enough universal for all of these research fields that have common goals: understanding the surrounding environment and designing decision-making process.

From our point of view, pervasive computing, ubiquitous computing, ambient intelligence are mostly focused on the smart home domain and healthcare. To the best of our knowledge, we did not find any references of pervasive computing, ubiquitous computing, ambient intelligence in other domains such as smart agriculture, transportation, etc. The main novelty of Internet of Things is to cover a wide range of application domains employing sensors and actuators. To achieve this, there is a need to provide interoperability among heterogeneous domains.

We synthesize for each SWoT related research field, their main features in Table 2.1

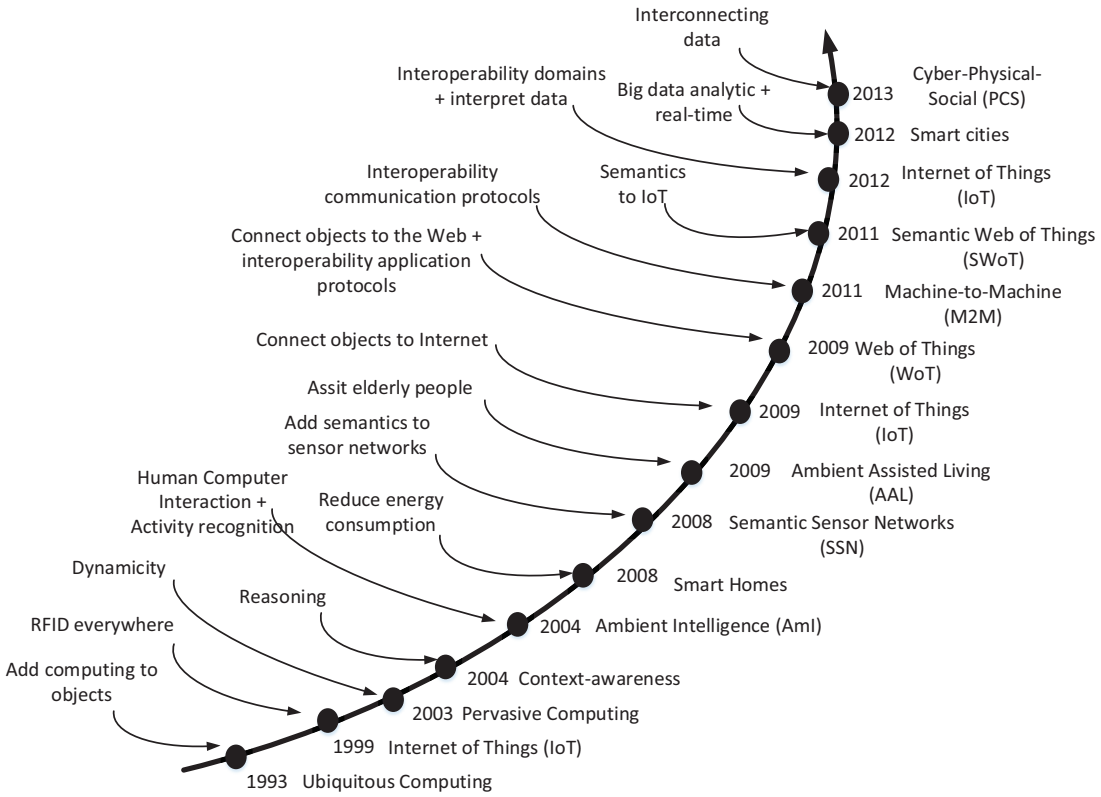

Figure 2.3: Evolution of Semantic Web of Things related research fields

and Figure 2.3. This is our point of view and we are agree that the table and picture are completely debatable, we indicate these dates and features according to the papers that we are familiar with and that we referenced in this chapter. For instance, the main research challenge in: (1) Machine-to-Machine (M2M) is to provide interoperability among network communication protocol, (2) Web of Things (WoT) is to focus on the interoperability between application protocols, and (3) Internet of Things (IoT) is to work on the interoperability between domains.

2.2 Identifying Main SWoT Challenges

We identified the following challenges as summarized in Table 2.2:

- **Interoperable IoT data** to ease sharing, reusing and combining sensor data. Related works are compared in 2.3.1.

Research fields / Challenges	Internet of Things (IoT)	Machine-to-Machine (M2M)	Web of Things (WoT)	Semantic Web of Things (SWoT)	Semantic Sensor Networks (SSN)
Interoperable IoT data	Ongoing	Ongoing	No	Yes	No. W3C SSN ontology is not enough
Interpreting IoT data	Ongoing	Ongoing	No	Yes	Yes but not easy to reuse.
Inter-domain interoperability	Ongoing	Ongoing	No	No	No
Designing interoperable SWoT applications	No	No	No	No	No
Securing IoT	Ongoing	Ongoing	No	No	No
Sensor Plug & Play	No	No	Yes	Yes	Yes
Adapted to constrained devices	Yes (CoAP)	Yes (CoAP)	Yes (CoAP)	Yes	Yes

Table 2.2: Main challenges of Semantic Web of Things and related research fields

- **Interpreting IoT data** to infer high level abstractions. Existing tools addressing this challenge are pointed out in 2.3.2.

- **Inter-domain interoperability** to ease the compatibility between heterogenous domains to build horizontal (e.g., cross-domain) IoT applications. This is explained in detail in section 2.3.3. Two sub-challenges are addressed: (1) reusing domain knowledge and (2) combining domain knowledge.

- **Designing interoperable SWoT applications** to assist users in developing IoT applications. The related works and tools regarding this challenge are explained in section 2.3.4.

- **Sensor Plug & Play** to automatically connect and recognize things based on heterogeneous hardware, software, operating system, etc. to get their data or order actuators. This challenge is addressed in section 2.3.5.
- **Semantics applied to constrained devices** to integrate semantic processing inside devices to avoid to send data to the cloud. Protocols and data formats should be lightweight and adapted to constrained devices. Further, the processing of IoT data should be feasible too. We explain this challenge in detail in section 2.3.6.
- **Securing IoT** applications and architectures to ensure security of sensor data. This is essential and requires securing communications and data. This challenge is addressed in section 2.3.7.

In Table 2.2, the legend is as follows: (1) 'Ongoing' means that the research fields are working on this challenge, but there is not concrete solutions yet, (2) 'No' means that research fields do not intent to address the problem, and (3) 'Yes' means that research fields work on these challenges and provide some concrete solutions such as tools.

2.3 Identifying Existing Tools Limitations for Each Challenge

In this section, we scrutinize related works and tools for each challenge: (1) interoperable IoT data, (2) interpreting IoT data, (3) inter-domain interoperability, (4) assisting IoT developer tasks, (5) 'Sensor Plug & Play' mechanisms, (6) semantics applied to constrained devices, and (7) security concerns for IoT.

2.3.1 Interoperable IoT Data

In United States, the Knoesis center designs two tools: SemSOS and Linked Sensor Data. **SemSOS** [Henson et al., 2009] has been designed for accessing and querying sensor data on the web. SemSOS uses the 52 North's SOS implementation[8] and enriches the SOS service with semantic annotations. It uses semantic web technologies to manage sensors and measurements. Patni et al. semantically annotate the MesoWest[9] weather dataset to generate the **'Linked Sensor Data'** [Patni et al., 2010a] [Patni et al., 2010b] [Pschorr et al., 2010] and 'Linked Observation Data' datasets available as Linked Data [Bizer et al., 2009]. They use the SWE standards to retrieve sensor measurements, convert data encoded with O&M into RDF, and then publish these semantic sensor datasets on the Web. These scalable datasets comprise 20,000 sensors, 160 million sensor observations and 1.7 billion RDF statements. Datasets have been enriched with contextual information using the GeoNames[10]

[8] `http://52north.org/communities/sensorweb/sos/`
[9] `http://mesowest.utah.edu/`
[10] `http://www.geonames.org/`

dataset to deduce regions, etc. They semantically annotate real sensor data for specific application purpose. They do not introduce the need to combine the domains with each other to provide smarter cross-domain applications.

In United Kingdom, the University of Surrey designs two tools: Sense and Sens'ability and Sense2Web tools. Regarding the **Sensei** project, [Barnaghi et al., 2009] develop a framework called **Sense and Sens'ability** which uses semantic technologies, the SWEET ontology for measurements units and SWE standards. **Sense2Web** is a Linked Data Platform for Semantic Sensor Networks based on an ontology called SensorData to publish semantic descriptions of sensors [Wei and Barnaghi, 2009] [Barnaghi et al., 2010] [Barnaghi et al., 2011] [De et al., 2012]. They introduce the need of using rule-based reasoning to deduce high-level abstraction from sensor data. In their recent works [De et al., 2012], they introduce the need to link measured data to domain knowledge through the Linked Data, but they do not provide a method to exploit and link domain ontologies relevant for IoT. They explain that they provide a M2M interface for publishing IoT data and associating it to existing vocabularies on the Web, but they do not explain how they do it. Recently, Barnaghi presented in a keynote given at Semantic Sensor Networks Workshop 2014[11], several main challenges: (1) how to make sense of sensor data, (2) application-dependant, (3) reusing domain knowledge, (4) semantics should be transparent to the end user, (5) discovery methods in the IoT and (6) provenance of sensor data. In this thesis, we are focusing on the four main challenges.

In Ireland, at the DERI/Insight institute, Le-Phuoc et al. develop the **SensorMasher** and the **Linked Stream Middleware (LSM)** platform to facilitate publishing 'Linked Stream Data' and making it available to other applications [Phuoc and Hauswirth, 2009]. They develop a user friendly interface to manage environmental semantic sensor networks.

In Spain, at the University of Madrid, Corcho et al. explain five challenges for the Semantic Sensor Web: (1) abstraction level from sensor data, (2) integration and fusion of data, (3) rapid development of applications, (4) identification and location of relevant sensor-based data sources, and (5) quality of service [Corcho and García-Castro, 2010]. In this thesis, we are focused on the tree former challenges. The **Semsor4grid4env**(Semantic Sensor Grids for Environmental Applications) project focuses on building large-scale semantic sensor networks for environmental management, in particular, for critical domain-specific scenarios such as fire prevention and flood control [Gray et al., 2011] .

In Australia, the CSIRO center works on 'Semantic Sensor Networks' too in the agricultural domain [Taylor et al., 2013] or ocean observations [Cameron et al., 2009]. **Kirby Smart Farm**[12] is based on Global Sensor Network (GSN) middleware [Aberer et al., 2006], SSN ontology and Linked Data. The *web of things* is developed in a farm [Gaire et al., 2013]. Cabral et al. develop a smart vineyard [Cabral et al., 2014].

Khusro et al. survey 'Linked Sensor Data' and underline the importance to semantically annotate sensor data to describe their meanings and make it potentially useful for future applications [Khusro et al., 2013].

[11] http://goo.gl/kjPHP7

[12] `http://smartfarm-ict.it.csiro.au/`

2.3.2 Interpreting IoT data

Interpreting sensor data is a time-consuming process. We should share the methods to avoid to reinvent the wheel each time. Finding, reusing and combining domain knowledge are challenging tasks.

In 2010, Moraru et al. use machine learning on sensor data [Moraru et al., 2010]. They use decision tree and Bayesian network to analyze their dataset comprising 16,578 measurements. They are focus on four kind of sensor measurements: temperature, humidity, light and pressure. Further, the dataset has additional information such as weekday, hour interval, position of the window, number of computers working and number of people in the lab. Then, in 2011, Moraru et al. propose a framework to enrich sensor data with semantics [Moraru, 2011] [Moraru and Mladenić, 2012] . They introduce the need of 'ontology collection' to provide context for sensor measurements, they use well-known ontologies: Geonames for location, Geo WGS84 for coordinates, the W3C SSN ontology to describe sensors, the SWEET ontologies and the W3C Time ontology. They do not mention the need of domain ontologies relevant for IoT. They explain as future directions the need of semantic reasoners to imply new knowledge from sensor data [Moraru, 2011]. In [Moraru et al., 2011], they describe the SemSense architecture to collect and then publish sensor data as Linked Data. In [Bradeško et al., 2012], the authors collect data on the fly and then validated and linked with Linked Open Data (LOD) datasets.

In 2012, Devaraju et al. design an ontology for weather events observed by sensors such as wind speed and visibility [Devaraju and Kauppinen, 2012]. They are focused on blizzard related phenomenon. They deduce high-level abstractions such as the types of snow (e.g., soft hail, snow, snow pellets), blizzard, winter storm, avalanche, flood, drought, and tornado. Such abstractions are deduced with rule-based reasoning, the implementation is based on SWRL and the Jess reasoning engine. They use the DUL ontology but the the W3C SSN ontology. They evaluate their approach with the Canadian Climate Archives database. Wang et al. explain that the SSN ontology "does not include modeling aspects for features of interest, units of measurement and domain knowledge that need to be associated with the sensor data to support autonomous data communication, efficient reasoning and decision making" [Wang et al., 2013] .

In 2014, Boshoven et al. interpret data produced by accelerometer, gyroscope, microphone, temperature and light sensors embedded in mobile phones [Boshoven and van Bommel, 2014]. They use the Hidden Markov Models and semantic web technologies to deduce activities. The rules are implemnted as SPARQL queries.

Ramparany et al. introduce the need of a domain-specific automated reasoning system [Ramparany et al., 2014]. It could be based on Description Logic or Complex Event Processing for interpreting IoT data. They do not propose a dataset with predefined rules that could be shared and reused by developers.

Roda et al. present a framework to support Intelligent Data Analysis (IDA) over sensor data by taking into account temporal abstractions [Roda and Musulin, 2014]. Their work is based on the W3C SSN ontology, DOLCE Ultra Lite (DUL) and SWRL Temporal ontology and develop a new ontology called Temporal Abstractions Ontology (TAO). They mention that they aligned these ontologies, but they do not align domain ontologies which can be used to infer high level abstractions neither show their interoperability issues. Their scenario

is a chemical plant.

In Table 2.3, we present heterogeneous approaches to enrich IoT data and we indicate pros and cons: (1) logic or rule-based reasoning, (2) machine learning, (3) Linked Stream processing, (4) reusing domain knowledge with Linked Open Data (LOD), Linked Open Vocabularies (LOV), Linked Open Rules (LOR). (5) distributed reasoning, and (6) recommender systems.

Methods	Pros	Cons
Logic/rule-based reasoning	- Simple rules - Adapted to simple sensors - Easy for beginners (learning & implementation) - Easier to combine rules	- Not adapted to complicated sensors - Heterogeneous rule languages and editors
Machine learning	- More elaborate results - Adapted to complicated sensors	- Need real datasets - Complicated for non-experts - Complicated for a 'sharing and reusing' approach
Linked Stream processing	- Real-time data - Scalability - Linked Data	- No real reasoning
Re-use domain knowledge (LOD, LOV, LOR)	- 'Sharing and reusing' approach	- Be familiar with semantics - Be familiar with ontology/instance matching tools - Not adapted to complicated sensors
Distributed Reasoning	- Scalability - Interoperability between systems	- Complicated for implementation
Recommendation systems	- Adapted to the user profile	- Complicated for non-experts - Need real datasets - Need user profile

Table 2.3: Existing approaches to enrich IoT data

In Table 2.4, we classify different tools according to the different approaches.

Semantic Perception, Inferring high-level abstraction

Henson et al. explain the idea of **'Semantic perception'** to interpret and reason on sensor data [Henson, 2013] [Henson et al., 2012] [Barnaghi et al., 2012a]. In his thesis, Henson develops an ontology of perception called IntellegO. He proposes a semantic-based approach to integrate abductive logic framework and Parsimonious Covering Theory (PCT) to interpret data. He explains that the development of background knowledge is a difficult task and out of the scope of his work. Therefore, there is a need to build a dataset which synthesizes domain knowledge expertise relevant for IoT which could be reused. Moreover, he outlines that perception does not enable a straightforward formalization using logic-based

Methods / Tools	Logic/rule-based reasoning	Machine learning	Linked Stream processing	Re-use domain knowledge (LOD, LOV, LOR)	Distributed Reasoning	Recommendation systems
'Semantic Sensor Web'	Yes	No	No	Yes	No	No
'Semantic Perception'	No	Yes	No	No	No	No
KAT	No	Yes	No	No	No	No
inContext-sensing	No	No	No	Yes	No	No
S-LOR	Yes	No	No	Yes	No	No
CQELS	No	No	Yes	Yes	No	No
SPARQLStream	No	No	Yes	Yes	No	No
WebPIE	No	No	No	No	Yes	No
DRAGO	No	No	No	No	Yes	No
Marvin	No	No	No	No	Yes	No
KaonP2P	No	No	No	No	Yes	No
LarKC	No	No	No	No	Yes	No

Table 2.4: Classification of tools according to reasoning approaches

reasoning. However, we do not completely agree with this fact as for simple sensors such as temperature or precipitation, logic-based reasoning is faster, flexible and easier for sharing. Regarding complex sensors such as accelerometer or ECG, this is true as logic-based reasoning is insufficient, and the use of data mining approaches is unavoidable.

Knowledge Acquisition Toolkit (KAT)

Ganz et al. design the **Knowledge Acquisition Toolkit (KAT)** to infer high-level abstractions from sensor data in gateways to reduce the traffic in network communications [Ganz et al., 2013] [Ganz et al., 2014] [Ganz, 2014]. KAT is composed of three components: (1) an extension of Symbolic Aggregate Approximation (SAX) algorithm, called SensorSAX, (2) abductive reasoning based on the Parsimonious Covering Theory (PCT), and (3) temporal and spatial reasoning. They use machine learning techniques (k-means clustering and Markov model methods) and then Semantic Web Rule Language (SWRL) rule-based systems to add labels to the abstractions. They propose to use domain-specific background knowledge, more precisely the Linked Data, but they do not clearly explain the incompatibility issues to reuse and combine the domain knowledge relevant for Internet of Things. They employ the abductive model rather than inductive of deductive approach to solve the incompleteness limitation due to missing observation information [Ganz, 2014]. Ganz et al. evaluate their work on real sensor data (temperature, light, sound, presence and power consumption). Their gateways support TinyOS, Contiki enabled devices and Oracle SunSpot nodes. Ganz et al. are focused on real-time processing of the data and high scalability

rather than assisting developers to design interoperable IoT applications [Ganz et al., 2015]. They explained that inferring high-level abstractions can be done through machine-learning techniques such as classification and clustering or even logical inference with the help of reasoning mechanisms and rule-based systems can be also used. They mentionned: 'the usage of domain ontologies increase the interoperability of data from different sources by applying a common model'. However, in practice, there are so many domain ontologies in the same domain (e.g., 45 ontologies in the smart home domain). It is hard to choose which ones to integrate in the system. Moreover, reusing domain ontologies implies new interoperability issues. They do not propose the idea to combine data from various domains or a dataset of unified semantic rules to share and reuse the way to interpret sensor data as it has been done for 'Linked Data'.

inContext-Sensing

The Spitfire project designed the **inContext-Sensing** tool. This tool enriches sensor data with the Linked Data by using the Pachube API, the SPITFIRE ontology and the Silk tool to align datasets such as DBPedia, WordNet, Musicbrainz, DBLP, flickrwrappr and Geonames [Leggieri et al., 2011]. The authors do not interpret the value of the produced sensor data by using domain-specific knowledge expertise.

Linked Stream Processing

Several works extend the SPARQL query language and engine to deal with stream sensor data and enrich them with the Linked Open Data cloud [Sequeda and Corcho, 2009]. C-SPARQL was an ealier proposal of streaming SPARQL system [Barbieri et al., 2009]. SPARQLStream has been proposed by Calbimonte et al. [Calbimonte et al., 2010] [Calbimonte, 2013] and CQELS by Le-Phuoc et al. [Le Phuoc, 2013] [Le-Phuoc et al., 2011]. In [Calbimonte, 2013], they design SPARQLstream, an extension to the SPARQL query language to deal with real-time sensor data. In [Le Phuoc, 2013], they design Continuous Query Evaluation over Linked Streams (CQELS) and combine data the the Linked Data.

Semantic-based distributed reasoning

In 2005, the **Distributed Reasoning Architecture for a Galaxy of Ontology (DRAGO)** distributed reasoning system, implemented as a peer-to-peer architecture, is designed by Serafini et al. [Serafini and Tamilin, 2005]. The goal of DRAGO is to reason on distributed ontologies. Kaonp2p has been designed by Haase et al. to query over distributed ontologies [Haase and Wang, 2007]. **LarKC (Large Knowledge Collider)** is a scalable distributed platform [Fensel et al., 2008]. **Marvin** is a scalable platform for parallel and distributing reasoning on RDF data [Oren et al., 2009]. Schlicht et al. propose a peer-to-peer reasoning for interlinking ontologies [Schlicht and Stuckenschmidt, 2010]. Taking inspiration from this work, there is a need to provide interoperable heterogeneous sensor-based rules and combine cross-domain ontologies and datasets in the context of IoT. In 2012, Abiteboul et al. see the Web as a distributed knowledge base and propose an automated reasoning over

this knowledge base [Abiteboul et al., 2012]. This work demonstrates the importance to reuse sensor-based domain ontologies and rules. In 2013, **WebPIE (Web-scale Parallel Inference Engine)** is an inference engine for semantic web reasoning (OWL and RDFS) based on the Hadoop platform designed by Urbani et al [Urbani et al., 2012]. WebPIE is scalable over 100 billion triples [Urbani, 2013]. Coppens et al. propose an extension to the SPARQL query language to support distributed and remote reasoning. For their implementation, they extend the Jena ARQ query engine [Coppens et al., 2013]. In 2014, Park et al. propose a semantic reasoning based on their XOntology and SPARQL. They use the Hadoop platform, HDFS and MapReduce to deal with thousands of sensor data nodes [Park et al., 2014].

None of these works, propose interoperable rules to interpret sensor data.

Cross-domain reasoning

Codina et al. propose a domain-independent recommendation system to provide personalization services for different domains (tourism, movies, books). They incorporate semantics into a content-based system to improve the flexibility and the quality, a domain-based inference (side-ward propagation, upward-propagation) for user's interests and a semantic similarity method to refine item-user matching algorithm [Codina and Ceccaroni, 2010b] [Codina and Ceccaroni, 2010a]. Hoxha et al. provide a cross-domain recommender system based on semantics and machine learning techniques (Markov logic) [Hoxha, 2014]. Tobias et al. provide a context-aware cross-domain recommender system. They exploit semantic web technologies and related tools such as DBpedia and the spreading activation algorithm citetobias2013semantic .

These works underline the importance of a cross-domain reasoning that we could also applied to sensor data.

2.3.3 Inter-Domain Interoperability

Schema.org[13] is a set of vocabularies to describe common data on the Web such as person, organizations, opening hours, places, etc. Major search engines and frameworks to build web applications agree on this set of vocabularies to facilitate the interpretation of data available on the Web and offer better search results to the users. Our goal is to use a similar approach applied to IoT to provide interoperable and understandable IoT data.

Reusing domain knowledge

In the semantic web community, numerous works preconize to reuse domain knowledge. However, existing tools to reuse domain knowledge that we studied do not enable to reuse the domain knowledge expertise relevant for IoT. Indeed, the domain knowledge is not referenced in semantic web search engines or catalogues since domain experts do not publish their ontologies, datasets or rules online and are not aware of semantic web guidelines.

[13] `http://schema.org/`

Borst et al. encourage the reuse of simple ontologies as they explained: "The domain knowledge must be carved up into modules containing different kinds of knowledge. This makes it possible to construct large and complex ontologies out of smaller and more reusable ones" [Borst, 1997].

Noy et al. explain in the second step of their ontology development methodology that ontology designers should consider reusing existing domain knowledge (e.g., ontologies) [Noy et al., 2001].

Bontas et al. explain that reusing ontologies enables: (1) implementation cost savings, (2) if the ontology is publicly accessible e.g., downloadable from a website, it can be used as input for the conceptualization phase, (3) tt increases interoperability when using the same ontology among various systems, and (4) building a domain ontology is inconceivable without a collaboration with domain experts [Bontas et al., 2005].

Stecher et al. clearly explain the benefits to reuse ontologies: (1) reduce the cost of creating ontologies, (2) improve the quality of the resulting ontologies, and (3) ease later interaction between systems [Stecher et al., 2008]. They distinguish three types of ontology-reuse: (1) conservative re-use, (2) adaptive re-use, and (3) best practice reuse. In this thesis, we are considering the adaptive re-use to exploit the domain expertise and the best practice reuse to exploit semantic web tools in later steps.

The **Neon** project[14] recommended the need to reuse available knowledge and proposed a set of methodologies [Suarez-Figueroa et al., 2012] [Suárez-Figueroa, 2010]. The authors of the Neon project focused on nine scenarios [Suarez-Figueroa et al., 2012]:

- Scenario 1: From specification to implementation
- Scenario 2: Reusing and re-engineering non-ontological resources
- Scenario 3: Reusing ontological resources
- Scenario 4: Reusing and re-engineering ontological resources
- Scenario 5: Reusing and merging ontological resources: ontology matching tools enable ontology aligning or merging.
- Scenario 6: Reusing merging, and re-engineering ontological resources
- Scenario 7: Reusing ontology design pattern (ODPs)
- Scenario 8: Restructuring ontological resources
- Scenario 9: Localizing ontological resources to translate of all the ontology terms into another natural language.

We are mainly interested in the scenario 3 to help IoT developers in reusing ontologies relevant for IoT. The others future steps are interesting for re-designing ontologies in an interoperable manner. 'Not reinventing the wheel at each ontology development' will speed up the ontology development process. They suggest to use semantic web search engines, but we have seen the limitations of such tools. Indeed, these tools do not take into account at

[14] `http://www.neon-project.org/`

all ontologies that are explained and described in research articles. Regarding the scenario 5, we analyze below ontology matching tool issues that we encountered (see section 2.3.3).

Semantic search engines such as Sindice[15] **[Tummarello et al., 2007], Watson**[16] **[d'Aquin and Motta, 2011] and Swoogle**[17] are not enough mature for finding domain ontologies relevant for IoT.

Datalift[18] [Scharffe et al., 2012] is a project to assist people in semantically annotating and linking data, but they are not focused on IoT and do not provide vocabularies related to IoT. The **Linked Open Vocabularies (LOV)**[19] [Vandenbussche et al., 2015] [Vandenbussche and Vatant, 2011] [Scharffe et al., 2012] [Vandenbussche et al., 2012] is a catalogue for ontologies, mainly known by semantic web experts. LOV lacks of ontologies relevant for IoT, and do not accept new ontologies if they do not follow their best practices. Unfortunately, such best practices are not known outside the semantic web community. The **DataHub**[20] is a catalogue for datasets. There is no quality checking when submitting a new dataset.

Euzenat et al. apply ontology matching to pervasive computing and focus to the dynamicity aspect [Euzenat et al., 2008]. They explicitly describe, page 11: "ontologies disseminated on the web provide the background knowledge necessary to interpret raw information". In practice, we have several challenging tasks: (1) semantically annotating the raw information provided by sensors, (2) finding the background knowledge fitting our needs, (3) getting the implementation of ontologies and rules, (4) choosing, employing ontology matching tools and interpreting the results, (5) integrating the reasoning mechanism, (6) designing the application exploiting the interpretation of sensor data. In practice, due to the heterogeneous nature of ontologies and rules and technical issues, even if they have been implemented with the same language (OWL, RDF, RDFS and SWRL), it is really difficult to reuse and combine the background knowledge. In [Euzenat et al., 2008], the authors do not recommend the standardization of ontologies since it will hamper the development of ontologies. We are not agree with this fact, for instance there are more than 44 ontologies to describe smart homes. We need to standardize at least a common basis to describe sensors and measurements used in smart homes. In the datalift platform, the authors assist users to semantically annotate data and linked it to other datasets, but not in the context of IoT [Scharffe et al., 2012]. Euzenat et al. provide tools for ontology matchings or alignments and the Linked Open Vocabularies (LOV), but not for the other tasks. They introduce the idea to assist developers in designing context-aware applications. There is a real need in assisting users with new tools to achieve these challenging and time-consuming tasks. Euzenat et al. apply ontology matching to the smart building domain and provide alignments to applications and devices [Euzenat, 2011]. Ontology matching is an intermediary stage since there is not yet standardized ontologies for smart homes. Scharffe et al. design the (Meta-Linking Data) MeLinDa framework to interlink the web of data [Scharffe and Euzenat, 2011]. They compare six tools according to different criteria:

[15] `http://sindice.com/`
[16] `http://watson.kmi.open.ac.uk/WatsonWUI/`
[17] `http://swoogle.umbc.edu/`
[18] `http://datalift.org/`
[19] `http://lov.okfn.org/dataset/lov/`
[20] `http://datahub.io/`

(1) degree of automation, (2) matching techniques used, (3) access to data, (4) ontologies exploited, (5) output produced, (6) domain-specific or not, and (7) post-processing. They mention the EDOAL (Expressive Declarative Ontology Alignment Language) language to share and reuse ontology alignments which are produced by matching tools.

Figure 2.5 shows the different tools (catalogue or semantic search engine) to encourage the reuse of domain knowledge and indicate pros and cons for each tool.

Descriptions / Tools	Feature	Pros	Cons
LOV	- Ontology catalogue	- More than 469 ontologies referenced - Ontologies designed by semantic web experts	- Not referenced if do not follow their requirements - Unsufficient for IoT - Semi-automatic
DataHub	- Dataset catalogue	- 9,195 datasets - Various formats accepted	- Unsufficient for IoT - No quality checked - Manually
READY4SmartCities	- Ontology & Dataset catalogue	- More than 50 ontology referenced	- Manually
LOV4IoT	- Domain knowledge relevant for IoT - Ontology, dataset & rule catalogue	- More than 200 projects referenced - Ontologies designed by domain experts	- Manually
Sindice, Watson, Swoogle	- Semantic search engines	- Automatic tools	- Unsufficient for IoT - Project not referenced if knowledge not available on the web

Table 2.5: Tools referencing domain knowledge

Nambi et al. propose the idea of an unified semantic knowledge base and is composed of several ontologies: (1) resource ontology to describe sensor, actuators or physical objects, which is an extension of the W3C SSN ontology, (2) location ontology extending the GeoNames ontology, (3) context & domain to represent contextual information and domain-specific knowledge, (4) policy, and (5) service [Nambi et al., 2014] .

Several works defend the need of reusing ontologies. Serrano et al. explain that it is important to reuse concepts designed in domain ontologies [Serrano et al., 2013]. The Jena Ontology API documentation[21] explains the same idea "an advantage of working with ontologies is that we can reuse work done by other ontologists".

[21] http://jena.apache.org/documentation/ontology/

Combining domain knowledge

Unifying the domain knowledge and terms is another challenging task. A naive approach is to integrate ontology matching tools [Kalfoglou and Schorlemmer, 2003] [Euzenat and Shvaiko, 2013] [Shvaiko and Euzenat, 2013] to achieve this task and answer the following questions:

- How to combine domain knowledge?
- Which ontology mapping should I use?
- Is the tool available online, easy to test and could be easily integrated in our framework?
- Does it contains a dictionary adapted to sensor measurements and IoT domain?
- Should we combine several mapping tools to achieve our task?

Khriyenko et al. explained the need of a common information or data model to reuse the data generated by devices to build new IoT applications, but do not mention the need to combine the data coming from various domains [Khriyenko et al., 2012]. They introduce the need of platforms for new IoT applications. They underline the need of ontology alignment for solving heterogeneity issues. They do not really test them with domain ontologies to see the results. They only explain the need of humans to control the results.

Recently, Manate et al. explained the need to employ domain-specific ontologies and ontologies matching and alignment tools to build IoT applications [Manate et al., 2014]. They underline that existing ontology mapping tools cannot execute with 100 percent accuracy for aligning or merging domain ontologies. They do not explicitly describe the issues encountered if we want to combine these domain ontologies. Firstly, ontology mapping tools are not enough mature for domain specific ontologies. Secondly, the domain ontologies have not been designed in a unified and interoperable way even with the use of standardized semantic web languages. They explained the need of four kind of IoT ontologies: (1) the sensor provenance ontology for identifying the data source designed by Patni et al. [Patni et al., 2010b], (2) the W3C SSN ontology ontology to describe sensor and actuators, (3) domain ontologies such as SWEET or DUL to add context to sensor data, and (4) service ontologies.

The existing ontology matching tools are not enough mature to combine domain ontologies relevant for IoT. There is not yet a unique tool addressing all of these problems in the same time:

- **Heterogeneous languages** means that the ontology can be written in various natural languages such as English, French, Chinese, Spanish or German. Detecting similar terms using different languages is harder for ontology matching tools. Some of the tools are based on the WordNet English dictionary, or EuroWordNet for European languages.
- **Syntactic heterogeneity** means that ontologies are implemented with different syntaxes and ontology editors tools. Ontology matching tools dealing with syntactic heterogeneity are not surveyed in the book written by Euzenat et al. (Chapter 2, page 39,

'This book is only concerned with reducing the terminological and conceptual types of heterogeneity') [Euzenat and Shvaiko, 2013].

- **Conceptual heterogeneity** means that a same concept can be represented in several ways in the ontology (e.g., concept or property) or in the dataset (e.g., instance). For instance, tired is defined as an instance or as a property as displayed in Figure 2.5. Further, ontologies for IoT are different from a structural point of view and best practices. For instance, concepts or properties do not have labels or comments whereas ontology matching tool algorithms are based on labels to compare them. Ontology Alignment Evaluation Initiative (OAEI)[22] is a benchmark frequently used to evaluate ontology matching tools, but their ontologies to match differ in their structure compared to the domain ontologies relevant for IoT (see Figure 2.4). Regarding ontologies relevant for IoT, concepts are linked with each other through `owl:Restriction`, and properties associated to concepts are not frequently used. For instance, snow is linked to temperature and precipitation through `owl:Restriction`. In the OAEI benchmark, concepts have properties which are mostly used by ontology matching tools. For instance, a person or a patient have both properties such as family name and birthdate.

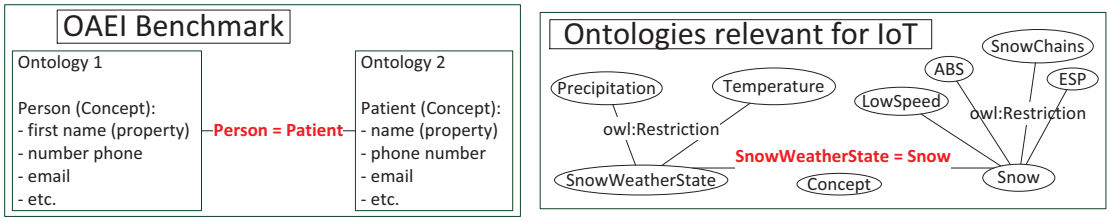

Figure 2.4: Comparison between OAEI benchmark and ontologies relevant for IoT

We detect the conceptual heterogeneity issue with LogMap[23] [Jiménez-Ruiz and Grau, 2011] where the authors explain us that LogMap will not do such matching. Another naive approach was to use the Silk platform[24] [Volz et al., 2009b] [Volz et al., 2009a] [Jentzsch et al., 2010] [Isele et al., 2010] for discovering, linking and maintaining data links between datasets.

Further, defining the threshold is difficult to avoid wrong results.

- **Terminological heterogeneity** means that different words are used to name the same entity such as equivalence (e.g., Snowy, SnowyWeatherState), etymology (e.g., fog/foggy) or synonyms. Even if ontology matching tools are based on a dictionary (e.g., Wordnet) and on a dictionary for synonyms (synset), this is not enough. These dictionaries are not sufficient for the IoT domain. Further, the classification of concepts can be difficult to detect. For instance, precipitation and rain are not considered as synonyms in WordNet but as hyponyms in Collins (see Figure 2.6). Dealing with

[22] `http://oaei.ontologymatching.org/`
[23] `http://csu6325.cs.ox.ac.uk/`
[24] `http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/`

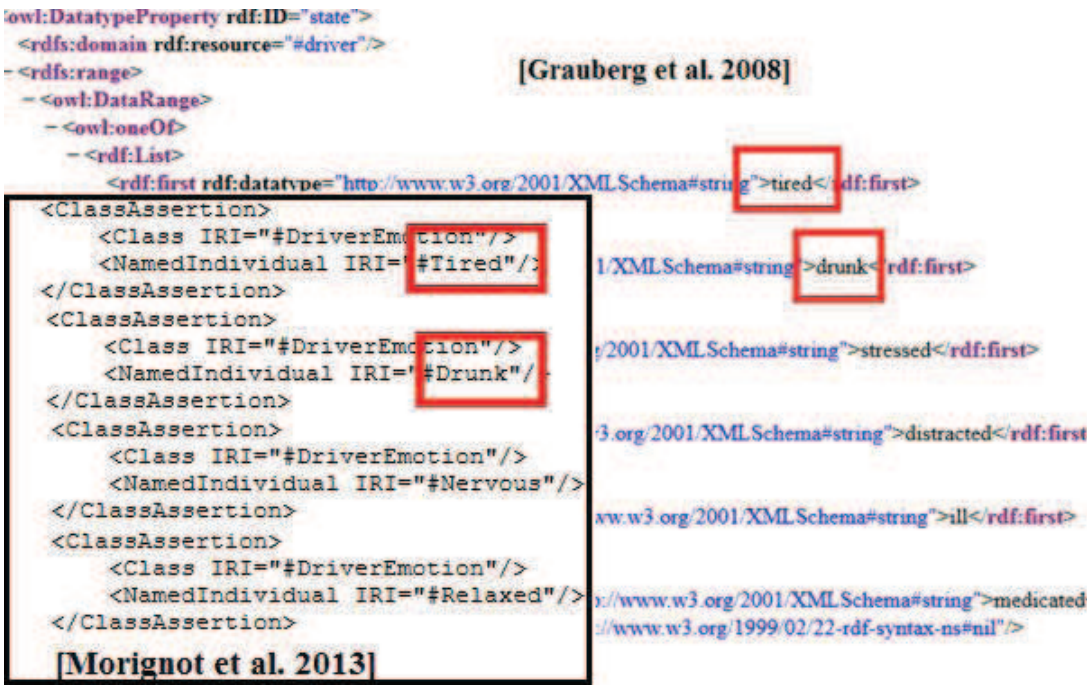

Figure 2.5: Ontology matching issues: entities not logical compatible

syntactic variations of entities such as abbreviations (Electrocardiogram and ECG) is not so easy for machines.

- **Semiotic heterogeneity** means that different entities are often interpreted by humans in different ways. For instance, how do you represent the concept 'orange', is it a fruit or an orange? If it is a fruit, do you want to explain that you can have several varieties of orange such as Naveline or Maltaise? Euzenat et al. did not survey ontology matching tools dealing with semiotic heterogeneity (Chapter 2, page 38, 'we do not deal with the semiotic heterogeneity here') [Euzenat and Shvaiko, 2013].

- Some tools are not implemented or available online. These tools cannot be tested to see if they fulfill our needs.

We used ontology matching LogMap [Jiménez-Ruiz and Grau, 2011], Aroma [David, 2007], Anchor-Prompt [Noy and Musen, 2001], MAFRA [Maedche et al., 2002] tools and for matching datasets, the Silk tool [Volz et al., 2009b] but the results are not those expected. For these reasons, we will not use ontology matching tools to combine domain knowledge.

2.3.4 Designing Interoperable IoT applications

Patel et al. describe the challenge to ease application development dedicated to smart office and fire management IoT applications [Patel et al., 2011] [Patel et al., 2013]. They propose

Figure 2.6: Precipitation and rain concepts are not defined in the same way (synonyms or hyponyms) in different dictionaries

a tool to easily develop IoT applications, but the application developers still need to program the application logic layer and they do not explain the way to interpret sensor data. They discuss the need of common domain vocabularies. However, their approach is not based on semantic web technologies and ontologies. They take into consideration actuators too. Their evaluation is based on two Eclipse plug-in: Metrics 1.3.6 and EclEmma to show that their tool reduce the development time. No demonstration is available and they do not provide end-user interactions.

Cassou et al. design a tool suite to develop pervasive computing applications and support the following stages: implementation, testing, and deployment [Cassou et al., 2012]. Their approach is not based on semantics. Adding new objects is less flexible since their code has to be modified and re-compiled. If the same objects were described in ontologies and datasets, the project will not need to be recompiled.

Paganelli et al. propose a similar idea to build a framework to speed up development of Web of Things applications based on web services such as REST but they do not propose

to interpret sensor data and link domains [Paganelli et al., 2014].

Ruta et al. propose a SWoT framework but not for reasoning on sensor data. They have several uses cases such as transportation and smart home [Ruta et al., 2012]. Further, they integrate semantics in constrained devices.

Hachem et al. explain the intervention of domain experts to interpret sensor data, which is costly and time-consuming [Hachem et al., 2011]. In her thesis, she explicitly describes as a long-term perspective the need to integrate inference mechanism to extract higher level knowledge from sensor data, since developers do not have the expertise for this task [Hachem, 2014]. This is exactly one of the goal of our proposed approach. She proposes 3 ontologies: the device ontology to represent physical things, the physics and mathematics domain ontology to provide formulas (e.g., speed=distance/time) and conversion for units (wind speed in km/h or mph) and the estimation ontology. She does not propose to reuse the domain knowledge that has already been integrated and interpreted in existing projects.

The ERLIoT (ErLang for the Internet of Things) framework assists developers in testing, debugging and verifying their code [Sivieri et al., 2014]. It is based on the Erlang programming language.

2.3.5 Sensor Plug & Play

Broring et al. provided a Sensor Plug & Play infrastructure based on SWE [Bröring et al., 2011b] and semantic web technologies. More specifically, they use the W3C SSS-XG ontology as well as the SWEET ontologies to represent environmental phenomena. They also use SWRL rules to define conversions such as simple unit or data type transformation. They provide a publish/subscribe mechanism to automatically: (1) recognize a new device in the system, (2) get the description of the new device, (3) publish measured data, and (4) retrieve the data. They also design a generic driver mechanism for sensors to exchange data between SWE services which avoids handling heterogeneous sensor protocols.

In the **Spitfire** project, they use ontology for the discovery mechanisms of sensors. They are not focused on the interpretation of sensor data to assist developers in designing SWoT applications [Pfisterer et al., 2011]. They introduce the need of automated integration and reasoning. Ishaq et al. focus on the automatic discovery of sensors. Their work is based on the CoAP protocol [Ishaq et al., 2012]. Mietz et al. explain the issue to represent sensor data in the IoT in an open way to efficiently combine them with other data [Mietz et al., 2013]. Further, they propose a reasoning based on Rule Interchange Format (RIF). However, to the best of our knowledge, no inference engine is available for this rule format. Further, if we want to reuse as much as possible the domain knowledge already implemented, most of the domain experts use the SWRL language since there are ontology editor tools and inference engine adapted to this language. In [Mietz et al., 2013], the authors use semantic-based prediction models to infer the high-level states of sensor values.

The **Global Sensor Network (GSN) middleware** eases a flexible integration and discovery of sensor networks and sensor data [Aberer et al., 2006]. XGSN is an extended version of GSN. Calbimonte et al. explicitly describe the step to infer high-level abstraction with domain-specific ontologies [Calbimonte et al., 2014]. They do not mentioned the interoperability issues when using domain-specific ontologies.

2.3.6 Semantics Applied to Constrained Devices

We report the related works on lightweight data format, integrating semantics and reasoning engine in constrained mobile devices in this section. They have been classified according to the following criteria: (1) provide a triple store, (2) a SPARQL query engine, and (3) a reasoning engine (see Table 2.6).

Descriptions / Tools	Triplestore	Query engine	Reasoning	Pros	Cons
RDF on the Go	Yes	Yes	No	- RDF storage (Based on Berkeley Database) - SPARQL query processor (Based on Jena ARQ)	-
microJena	No	No	Yes	- Recommended by Apache - Reasoning supported	- SPARQL not supported - SWRL not supported
AndroJena	Yes	Yes	Yes	- Easy to use - Reasoning supported - SPARQL supported (ARQoid) - Triplestore (TBoid)	-
Otsopack	Yes	No	No	-	-
microOR	No	No	Yes	-	-
Delta-Reasoner	No	No	Yes	-	-
ELK	No	No	Yes	-	-

Table 2.6: Semantic tools for constrained devices

Lightweight protocol and data format

To retrieve and describe data generated by M2M devices, there are two main possibilities: Sensor Markup Language (SenML) and Sensor Web Enablement (SWE) standards . Both protocols bridge the gap of interoperability of M2M data. Existing protocols and data formats employed are mostly proprietary. Most of the 'Semantic Sensor Networks' works are based on Sensor Web Enablement (SWE).

Sensor Web Enablement (SWE) is not adapted to constrained devices such as mobile phones [Botts et al., 2008] [Bröring et al., 2011a]. Further, SWE does not provide mechanisms to support the description of actuators and RFID tags measurements. Aggarwal et al. discuss the challenge to interpret sensor data. They say: "too much data, too little interoperability and too little knowledge" [Aggarwal et al., 2013] . They recommend SWE and not SenML as a format for sensor data.

Sensor Markup Language (SenML) is a low-energy consuming language to retrieve measurements from M2M devices and a non-proprietary format [Arkko, 2012] . SenML

provides simple measurements: the name, the units, and the value. For example, a measurement can be the temperature, the value 5, and the units degree Celsius. SenML is a lightweight protocol to get simple measurements but does not provide services such as proposed by SWE. Recently, Su et al. convert senML to RDF for the fishery domain, but do not provide any tools available online [Su et al., 2014b]. In [Su et al., 2014a], the authors compare various SenML data format used to represent sensor measurements in constrained devices such as JSON, XML, EXI. SenML is used to describe sensor metadata in a simple way and has been implemented in [Datta et al., 2014b]. However, there is a need to semantically annotate the SenML measurements directly in constrained devices to interpret sensor data. **Constrained Application Protocol (CoAP)** [Bormann et al., 2012] is used to get access to sensor data between constrained devices. Recent works are based on CoAP and SenML. **Constrained RESTful Environments (CoRE) Link Format** [Shelby et al., 2012] is used to have REST web services suitable for constrained devices.

Integrating semantics in constrained devices

There are several works related to integration of semantics into mobile devices.

Le-Phuoc et al. designed **RDF On The Go**, a RDF storage and SPARQL query processor for Android mobile phones [Le Phuoc et al., 2010]. They use Jena framework and ARQ Semantic Web Toolkit to execute SPARQL queries and lightweight Berkeley database for storing the RDF data. Their demonstration shows ten nearest cafes or fast food restaurants to the current location obtained from the GPS co-ordinates. This approach reduces network operations and limits the amount of user information exposed to remote server thereby preserving privacy.

microJena is referenced by Apache Jena and provides a way to load and manipulate RDF data on a device, a reasoning engine optimized for resource-constrained devices. However, SPARQL queries and SWRL rules are not supported.

AndroJena is a light version of the Jena framework to build Smeantic Web applications. To overcome technical issues, AndroJena has beed adapted to Android devices. AndroJena provides several extensions: (1) the reasoning engine, (2) the ARQoid SPARQL engine, and (3) the TDBoid triplestore. Although this tool is not recommended by Apache, it fits our needs since it enables both reasoning and querying in Android-based constrained devices. David et al. use AndroJena to integrate semantics in mobile phones [David et al., 2012a]. Further, they use ProGuard to delete the unused classes, methods or variables to optimize the code. They provide a mobile API for Linked Data [David et al., 2012b].

MobileRDF is a lightweight RDF API. **Otsopack** is a triple store for devices with limited computational resources [Gómez-Goiri et al., 2014]. **microOR** is an ontological reasoner optimized for resource constrained devices [Ali and Kiefer, 2009].

Reasoning engines for mobile devices

Delta-Reasoner is a reasoning engine developed for constrained mobile devices [Motik et al., 2012]. The engine is embedded into context-aware applications which can interpret the current situation of the mobile device user based on different sensor measurements. The sensors

considered may include internal sensors (e.g. GPS), external sensors (e.g. indoor location) and pseudo sensors. There is no implementation of this tool available online.

Another reasoner called **ELK** is designed for lightweight OWL EL [Kazakov et al., 2012]. The authors describe the algorithm which supports high performance reasoning in mobile devices and demonstrated it on a Google Nexus 4 running Android 4.2.

Ontology-based mobile applications

Becker et al. developed the **DBpedia mobile**, that queries a semantic Wikipedia, through a mobile browser [Becker and Bizer, 2008]. DBpedia mobile exploits user's location information and a linked data browser to propose nearby tourist attractions.

Ruta et al. propose **iDriveSafe 2.0**, an ontology-based application on mobile phones in transportation domain [Ruta et al., 2010] . The primary goal of the application is to display vehicle health (emission, fuel consumption, gear level). It can also suggest the use of safety devices according to the weather conditions (e.g., switch on the fog lamp if the weather is foggy) and detect the driver's state (careful, aggressive, tired).

D'Aquin et al. propose the **SmartProducts project** that is compliant with W3C SSN ontology and is composed of ontologies related to food and recipes [d'Aquin et al., 2011]. They enable mobile applications to expose data though a SPARQL endpoint on Android devices [d'Aquin et al., 2011]. They integrate a Sesame triple store in Android to store data in the SDCard, and the iJetty server to be compatible with Android.

An ontology-based mobile application for healthcare applications is discussed [Liu, 2013]. The work presents an in-depth discussion on semantic interoperability and uses the Shimmer sensor for acceleration data and the Philips DTI-2 sensor for skin conductance and acceleration measurements. But the resulting application developed just displays data on the mobile phones, no rules or suggestions are provided.

Chien et al. propose a tourism ontology to provide a museum/exhibit-guidance system [Chien et al., 2013].

Skillen et al. develop an ontology-based Android application to address the problem of the outdoor mobility of an elderly person such as shopping trip reminders or context-aware guidance [Skillen et al., 2012].

2.3.7 Securing IoT

In this section, we are focus on security ontologies that can be useful to secure IoT applications, ETSI M2M and oneM2M architectures. We briefly introduce main security challenges in IoT and M2M. Then, we present several security ontologies.

OWASP IoT[25] explains the need to secure IoT by classifying the most 10 frequent vulnerabilities: insecure web interface, insufficient authentication/authorization, insecure network services, lack of transport encryption, privacy concerns, insecure cloud interface, insecure mobile interface, insufficient security configurability, insecure software/firmware, and poor physical security.

[25] `http://goo.gl/dTEbLF`

OneM2M [OneM2M and Security, 2014] analyzes security threats which may arise in the OneM2M architecture and the related countermeasures. In OneM2M [OneM2M and Security, 2014], the authors explain the importance of securing M2M applications and network communications. Borgohain et al. survey the main security issues specific to wireless sensor networks and RFID [Borgohain et al., 2015] . They do not introduce security issues for other technologies involved in IoT or M2M such as cellular technologies, web applications, etc. The survey is incomplete. Chen et al. explain privacy protection mechanisms to secure M2M devices, M2M communications, storage and processing of M2M data [Cheng et al., 2012].

Bandyopadhyay et al. underline the need to secure IoT architectures at design phase [Bandyopadhyay and Sen, 2011].

Indeed, Alam et al. underline the need of security reasoning for IoT through ontologies and semantic rules [Alam et al., 2011]. They outlines security requirements for IoT such as confidentiality, integrity, authentication, authorization, access control, trustworthiness, etc. Confidentiality should be achieved to secure sensitive data through encryption. Integrity is required to check that sensed, stored and transmitted data have not been tampered either maliciously or accidently. They do not suggest which security mechanisms we should integrate in our own IoT applications.

Securing IoT devices

In this section, we present the security ontologies related to M2M devices, more precisely, sensors, embedded sensors and mobile phones:

- **Security Ontologies for Sensor Networks:** We found only two ontologies defining the security concepts for Wireless Sensor Networks. Znaidi et al. propose an ontology defining only the classification of attacks according to the OSI model [Znaidi et al., 2008]. They neither describe well-known attacks specific to the transport layer such as desynchronisation, DoS and flooding nor security mechanisms, protocols and key management specific to sensor networks. Kenfack et al. define intrusions in wireless sensor networks [Kenfack et al., 2011]. They classify vulnerabilities, such as shared wireless medium, lack of infrastructure and easy physical accessibility by the intruders and describe WSNs components (e.g., battery, sensor, radio). The main shortcomings of this work is that, firstly, none of these ontologies mention sensor security mechanisms and security properties, and secondly, these ontologies are not published online.

- **Security Ontologies for Mobile Phones:** Beji et al. propose a security ontology for mobile applications divided in three sub-ontologies: (1) The Asset-Vulnerability-Threat Ontology (AVTO) to classify the vulnerabilities into three main classes: physical, software and those related to communications, (2) the Mobile Profile Ontology (MPO) and (3) the Defense Mechanism Ontology (DMO) which describes the main security and cryptographic mechanisms such as digital signature, locking mechanism, encryption, key management, PKI, access control methods, algorithm and those specific to the mobile field (SIM locking) [Beji and El Kadhi, 2009]. Vincent et al. define an ontology-based firewall to ensure privacy protection for smartphones. They propose two ontologies: (1) privacy policies designed with SWRL, inspired by the SOUPA

framework, and (2) the digital identity on smartphones using well-known ontologies FOAF and VCard. None of these ontologies are published online [Vincent et al., 2012] [Vincent et al., 2011a] [Vincent et al., 2011b].

Securing IoT network communications

Several security ontologies have been defined for network communications, more precisely, cellular networks (2G, 3G, 4G) and Wi-Fi. These ontologies mainly describe security mechanisms in the physical and link OSI model layer:

- **Security Ontologies for Cellular Networks:** Three ontologies describing the architecture of cellular networks and the associated security mechanisms have been designed: Long Term Evolution (LTE)/4G [Neji and Bouallegue, 2012a], Universal Mobile Telecommunications System (UMTS)/3G [Neji and Bouallegue, 2012b] and Global System for Mobile Communication (GSM)/2G [Neji and Bouallegue, 2012c]. Alazeib et al. have developed an ontology to describe GSM, UMTS and wireless Local Area Network (WLAN) network architectures [Alazeib and Diehl, 2005]. This ontology defines also the authentication mechanisms applied to these technologies.

- **Security Ontologies for Intrusion Detection Systems:** Joshi et al. have designed the Intrusion Detection System ontology with classes such as Vulnerability, Product, Attack properties and Weakness [Joshi, 2013] [Undercoffer et al., 2003]. This ontology is used to convert the National Vulnerability database (NVD) into RDF and is compliant with Linked Data principle but not Linked Open Vocabularies principles. Tsoumas et al. define an ontology for security mechanisms such as firewall, antivirus and network protocols [Tsoumas et al., 2005]. Frye et al. define the attack ontology to identify complex network attacks [Frye et al., 2012] and Salahi et al. an ontology to predict networks attacks [Salahi and Ansarinia, 2013].

Securing IoT applications and IoT data

Several security ontologies describing cryptographic concepts and usual security mechanisms have been defined to secure M2M data.

- **General Security Ontologies:** Souag et al. review numerous security ontologies and underline that they are not published online but they do not explain that most of the existing works do not follow the semantic web best practices [Souag et al., 2012]. Kim et al. created seven ontologies. The main security ontology describes security concepts such as security objectives (e.g. authentication) and network security protocols (e.g., IPSec, SSL) [Kim et al., 2005]. Another ontology describes symmetric and asymmetric algorithms, hash algorithms, key exchange algorithms and digital signatures. Herzog et al. propose four ontologies defining several concepts such as assets, threats and vulnerabilities [Herzog et al., 2007]. They also define concepts for: (1) security mechanisms such as asymmetric and symmetric algorithms that are

classified into block cipher or stream cipher, (2) secure network communication protocols such as SSL, SSH, VPN, (3) security goals (authentication, integrity, confidentiality), and (4) access control model (RBAC, MAC, DAC). Denker et al. define two ontologies called 'security mechanisms' and 'credential' [Denker et al., 2004] [Denker et al., 2003]. They propose the notion of security notations to represent security properties such as authentication or confidentiality. They also define concepts for different authentication methods: certificate-based, password-based, biometrics (fingerprints, voice) and physical components (e.g., card). Lekhchine et al. propose an ontology called Mobile Agent Security Ontology (MASO). This ontology is written in French which defines concepts for symmetric/asymmetric algorithms, hash function, security goals and security mechanisms such as firewall and antivirus [Lekhchine, 2009]. Vorobiev et al. define several ontologies: (1) Security Attack Ontology (SAO), (2) Security Defence Ontology (SDO), (3) Security Asset-Vulnerability Ontology (SAVO), (4) Security Algorithm-Standard Ontology (SASO), and (5) Security Function Ontology (SFO) [Vorobiev and Bekmamedova, 2010]. Evesti et al. designed an ontology to describe and check the age or structure of the password and the authentication level [Evesti et al., 2011]. Villata et al. design the Social Semantic SPARQL Security for Access Control Ontology (S4AC) to provide access control over linked data or social web [Villata et al., 2011a] [Villata et al., 2011b]. This work could be reused and adapted to IoT to provide access control on sensor data. For instance, usually, health data need to be protected compared to weather data. Costabello's thesis aims at enhancing Linked Data access for context-awareness [Costabello, 2013]. The author designs the PRISSMA lightweight context ontology which is exploited in the PRISSMA framework and in the Shi3ld system, a context-dependent authorization based on the S4AC ontology.

- **Security Ontologies for Web Applications:** Ekelhart et al. propose the AURUM framework, an ontology-based security knowledge [Ekelhart et al., 2009]. They do not classify the security mechanisms and attacks according to the technologies. Razzaq et al. define an ontology to classify web application attacks such as cookie poisoning, SQL injection, Cross Site Scripting (XSS) and design SWRL rules [Razzaq et al., 2014]. Finally, Huang et al. propose an ontology-based malware behavioral analysis, called Taiwan Malware Analysis Net (TWMAN), that focus on malware ontology with concepts such as trojan, backdoor, worm [Huang et al., 2010].

2.4 Concluding Remarks: Limitations of these Works

We deduce from this state of the art five main challenges that we describe below: (1) interoperable IoT/M2M data, (2) interpreting IoT/M2M data, (3) inter-domain interoperability, (4) designing interoperable IoT/M2M applications, and (5) securing IoT/M2M.

All of these challenges highlight the necessity to assist developers in designing secure interoperable cross-domain Semantic Web of Things (SWoT) applications. We do not address in this thesis the two other challenges that we highlighted: 'Sensor Plug & Play' and semantics adapted to constrained devices since many projects are already involved in filling these gaps.

We recapitulate in table 2.7 the main shortcomings of the current standardization works conducted by working groups such as ETSI M2M [M2M, 2012], oneM2M [OneM2M et al., 2014], W3C SSN ontology[26] [Compton et al., 2012] and W3C Web of Things[27]. Table 2.8 summarizes limitations of the main existing Semantic Web of Things frameworks.

Challenges / Standards	W3C SSN ontology	ETSI M2M	oneM2M	W3C Web of Things
Interoperable IoT data	No ("Do not provide a basis for reasoning") They focused on interoperable sensor networks	No	Ongoing (Semantic annotation to enable automated processing of semantic information)	No
Interpreting IoT data	No ("Do not provide a basis for reasoning")	Ongoing (Make use of the data)	Ongoing (Semantic annotation to enable automated processing of semantic information)	Ongoing (Interpreting sensor input)
Inter-domain interoperability + Reusing domain knowledge	No ("Does not describe domain concepts"). No interoperability among domain ontologies No (Do not introduce ontologies relevant for IoT)	Ongoing (Stays independent from "vertical markets", application-specific. Re-use of M2M data across different applications). (Reuse of data not necessary the domain knowledge)	Ongoing (Interoperability between "siloed" applications, data interoperability to heterogeneous M2M applications) (Extensible ontologies with new domain concepts)	Ongoing (What is needed to encourage use of common vocabularies and how should these be standardized?)
Designing interoperable SWoT applications	No	Ongoing Interoperable M2M applications	Ongoing Interoperable M2M applications	No
Securing IoT	No	Ongoing	Ongoing	Ongoing

Table 2.7: Limitations of current standardizations and working groups

[26] http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/
[27] `http://www.w3.org/2014/02/wot/`

Projects / Challenges	Semantic Sensor Web (Henson, Sheth, Patni et al.) 2008-2014	CityPulse (Barnaghi, Ganz, Wang et al.) 2009-2014	SWoT (Ruta et al.) 2010-2012	Spitfire 2011-2013
Interoperable IoT data	Yes but not enough (SSN ontology, Linked Sensor Data)	No (SSN ontology not enough)	No	No (SSN ontology not enough)
Interpreting IoT data	Yes. ("Semantic Perception" with abductive logic PCT)	Yes. ("High level abstraction", KAT tool)	No	No. inContext tool to link datasets not interpret values
Inter-domain interoperability + Reusing domain knowledge	No (Scenarios: health + weather)	No (Scenarios: home + weather)	No (Scenarios: home + transport)	No
Designing interoperable IoT applications	No	No	No	No (not for application level)
Securing IoT	No	No	No	No

Table 2.8: Limitations of SWoT frameworks

2.4.1 Describing interoperable IoT data

More and more projects use semantic web technologies to ease interoperability. Few of them focus on semantically annotating IoT data. Recent works [Jara et al., 2014] introduce the need to describe sensor data in an interoperable manner.

Defining a nomenclature with common terms to describe sensor measurements to later easily interpret and combine them is becoming essential.

2.4.2 Interpreting IoT Data

'Semantic Sensor Networks' works are mainly based on machine learning methods to interpret data and reuse popular vocabularies such as W3C Time, Geonames, DBPedia but

not domain-specific vocabularies relevant for IoT such as smart home, healthcare or agricultural ontologies. However, most of the other related research fields presented above employed rule-based reasoning mechanisms based on Semantic Web Rule Language (SWRL) [Horrocks et al., 2004]. Since, SWRL rules are increasingly popular and used, we would like to combine and reuse the existing rules already designed previously. Moreover, frequently, these rules are interconnected with ontologies that we could reuse too. To the best of our knowledge, existing SWOT-related projects, standardizations or working groups do not provide any concrete solutions to interpret sensor data that we can easily share and reuse. Further, most of the works do not propose to reuse the domain knowledge expertise.

There is a real need to find approaches to easily share and reuse the way to interpret sensor data and reuse the domain knowledge expertise already designed. We could design a dataset of interoperable ontologies and rules to reason on sensor data. Such approaches will be based on 'Linked Open Data' approaches.

Some previous works use the term 'domain ontologies' to add context to sensor data by using ontologies such as W3C Time to add temporal context. In the remainder of this thesis, we will use the term 'domain ontologies' to describe ontologies relevant for IoT such as smart home or healthcare ontologies which could be reused to interpret sensor data.

2.4.3 Inter-domain Interoperability

ETSI M2M [M2M, 2012] and oneM2M [OneM2M et al., 2014] standardization bodies emphasize the need to combine domains but do not provide any methods. Existing works do not provide an interoperable domain knowledge to easily build cross-domain IoT applications. To achieve this task, there are two sub-challenges:

- Reusing domain knowledge: Semantic web tools do not reference domain knowledge relevant for IoT. Most of the domain ontologies are: (1) not published online, (2) do not follow semantic web best practices, and (3) are not interlinked. For these reasons, most of the existing works do not exploit the domain knowledge expertise already designed and implemented.

- Combining domain knowledge: From a technical point of view, it is a challenging task to deal with heterogeneous syntaxes due to the use of various ontology or rule editors and reasoning engines. Even with the use of semantic web standardized languages, this knowledge is not interoperable. Finding the most suitable ontology matching fitting specific needs is really time-consuming and do not fulfill our needs.

There is a necessity to build a dataset to reference and classify domain ontologies relevant for IoT to encourage people to reuse the domain knowledge. Based on this dataset, we will redesign an interoperable domain knowledge. Such steps will enable to assist users in designing interoperable cross-domain IoT applications.

2.4.4 Securing IoT

Existing approaches introduce the need of securing IoT applications and architectures to assist developer tasks but do not provide any concrete solutions. More and more works are based on security ontologies, but still have shortcomings:

- Not designed for IoT and M2M.
- Lack of unified terms: the main drawback of existing ontologies is that they use different names for the same concepts which can confuse a software developer which is not expert in security. For example, we frequently found several terms such as goal, security notation, security objective in these ontologies for defining the 'security property' concept (e.g., confidentiality).
- Incomplete security knowledge: most of these ontologies are domain specific. There is a need of a security knowledge base to: (1) classify both threats and security mechanisms according to various technologies, (2) classify attacks and security mechanisms according to the OSI model, (3) describe strengths and weaknesses of security mechanisms, and (4) specify the relationships between security mechanisms, attacks and security properties.
- Lack of semantic web best practices: frequently, security ontologies cannot be reused since they are not published online, referenced on semantic web tools or do not follow the semantic web best practices. They are not interlinked with each other whereas they design similar concepts.

There is a need to assist developers in finding the right security mechanisms to secure their architectures or applications. A solution is to design an unified security knowledge base to assist them in finding security mechanisms fitting their needs.

2.4.5 Summary

As said previously, based on the limitations and shortcomings of existing works, five main challenges must be addressed. In Part II, we will show how these challenges have been solved in the context of this thesis:

- Challenge A: Interoperable IoT data will be addressed in Chapter 4.
- Challenge B: Interpreting IoT data will be addressed in Chapter 4.
- Challenge C: Inter-domain interoperability will be addressed in Chapter 4.
- Challenge D: Designing interoperable SWoT applications to assist IoT developers will be overcome in Chapter 3.
- Challenge E: Securing IoT applications will be addressed in Chapter 5.

In the next chapter (Chapter 3), we will present in detail the innovative approach called Machine-to-Machine Measurement (M3) that we designed to maximize the productivity of SWoT application developers for: (1) interpreting interoperable IoT data, (2) ensuring inter-domain interoperability, and (3) taking into consideration security concerns.

Part II

Contributions

In Chapter 3, we describe our first contribution: the Machine-to-Machine Measurement M3 framework that addresses the challenge of designing SWoT applications. In Chapter 4, we present the second contribution namely the Sensor-Based Linked Open Rules S-LOR that addresses three challenges: (1) providing interoperable IoT data, (2) interpreting IoT data, and (3) reusing domain knowledge expertise. Finally, in Chapter 5, we detail the third contribution the Security Toolbox: Attacks & Countermeasures STAC designed to secure IoT applications.

Chapter 3

The Machine-to-Machine Measurement (M3) Framework

"Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But its worth it in the end because once you get there, you can move mountains."

Steve jobs

"If I had asked people what they wanted, they would have said faster horses."

Henry Ford

In this chapter, we assist IoT developers tasks in addressing "Challenge D: Designing interoperable Semantic Web of Things (SWoT) applications". We assume in this work that the developers want to design IoT applications to interpret sensor data. Constantly, IoT developers accomplish four tasks as depicted in Figure 3.1: (1) design SWoT applications, (2) semantically annotate IoT data, (3) interpret IoT data, and (4) secure IoT applications. Developers constantly learn semantic web technologies and tools to design their own solutions, which is really time-consuming and neither reusable nor interoperable. In all of these steps, our goal is to help them as much as possible. To solve this challenge, we conceive the Machine-to-Machine Measurement (M3) framework to assist developers in designing interoperable and cross-domain SWoT applications. M3 will automatically generate the code for the fourth tasks that the developers will have to achieve. The main added value of the M3 framework is to enable developers designing Semantic Web of Things applications without learning semantic web technologies. Further, it provides time-saving and interoperable SWoT applications, even if they have been developed by two distinct developers.

This chapter comprises the following sections. Section 3.1 introduces high-level M3 components to assist developers in designing interoperable SWoT applications. Section 3.2 presents the M3 architectural overview. Section 3.4 describes how developers can exploit

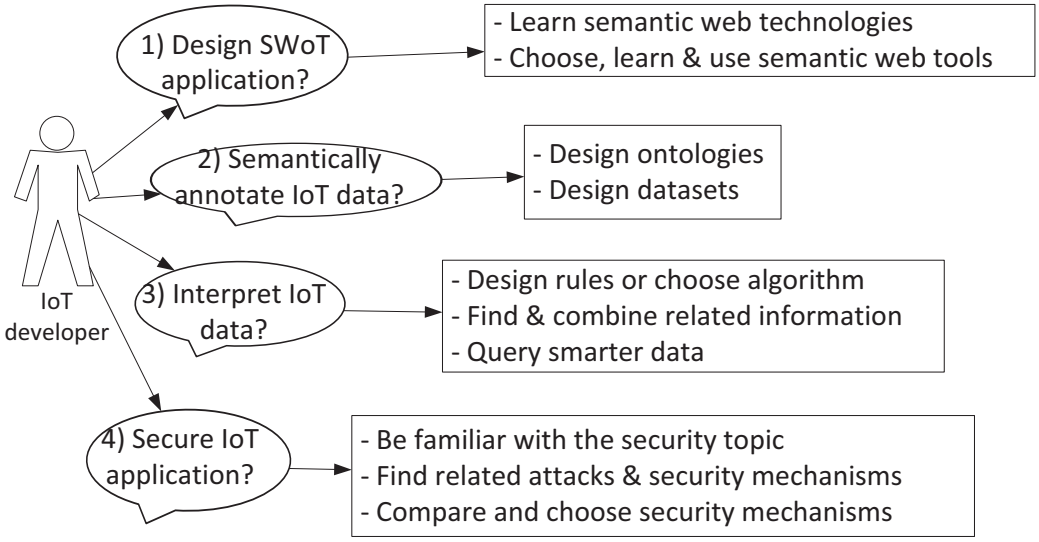

Figure 3.1: Time-consuming tasks performed by IoT developers

the M3 framework to design SWoT applications through M3 web services or a user interface. Section 3.5, demonstrates that our proposed framework has been integrated in a semantic-based ETSI M2M architecture. Section 3.6 provides a proof-of concept. Section 3.7 evaluates the M3 framework. Finally, section 3.8 concludes this chapter and outlines future work.

Thanks to M3, we assess the following research questions:

- How to assist developers in designing SWoT applications?
- How to generate interoperable domain-specific or cross-domain SWoT applications?

3.1 Assisting Developers in Designing SWoT applications

The main objective of the M3 framework is to assist developers in designing SWoT applications to easily interpret IoT data. We mentioned that IoT developers accomplish four tasks as depicted in Figure 3.1. For each of these tasks, we designed a tool to help them: (1) design SWoT applications with the Semantic Web of Things (SWoT) generator, (2) semantically annotate IoT data with the M3 converter, (3) interpret IoT data with Sensor-based Linked Open Rules (S-LOR), and (4) secure IoT applications with Security Toolbox: Attacks & Countermeasures (STAC).

In Figure 3.2, in the first step, developers get a set of files composed of reusable domain knowledge that we called 'M3 template' through the **SWoT generator** by providing sensors and domains used. In this figure, the developer is the same person (just for visibility reasons, we duplicate the picture of the developer). The template avoids to the developers

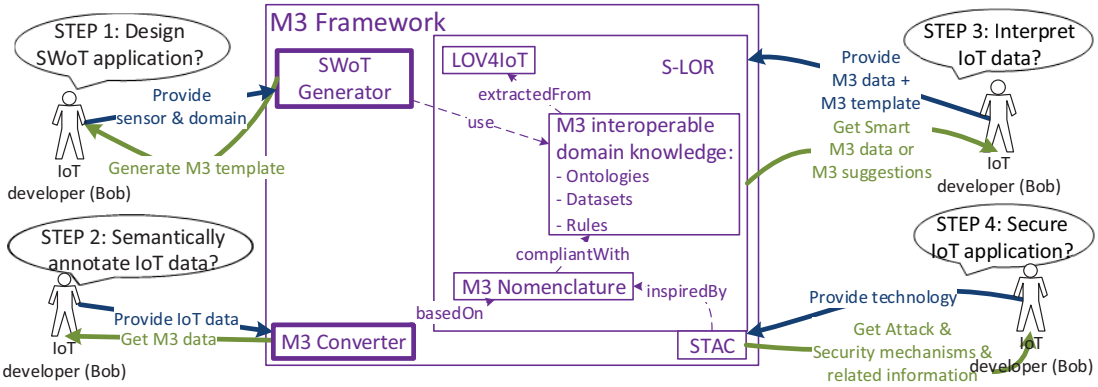

Figure 3.2: M3 assists developers in designing SWoT applications

to design its own ontologies, datasets and rules. Then, in the second step, IoT developers send IoT data to the **M3 converter** which semantically annotates it according to the **M3 nomenclature** to get interoperable M3 data. Then, in the third step, they easily interpret M3 data and enrich it with the **M3 interoperable domain knowledge** provided by the M3 template. Thanks to this approach, the developer task is focused on parsing and displaying M3 suggestions in an user-friendly interface. The developers could even send notifications or control actuators (e.g., close the door). Finally, if required, in the fourth step, the developers can be assisted by **STAC** to secure their IoT applications. STAC suggests the security mechanisms to integrate in their applications based on the technologies employed. All of the M3 components are briefly explained in Table 3.1.

M3 Framework Component	Description
SWoT Generator	Generates M3 templates (ontology, rules, datasets and SPARQL query) to easily design SWoT applications.
M3 Converter	Semantically annotates IoT data according to the M3 nomenclature.
M3 Nomenclature	Describes IoT data in an interoperable manner to ease reasoning. An explicit context is also added. The M3 nomenclature is implemented in the M3 ontology.
LOV4IoT	A knowledge base relevant for IoT used to enrich IoT data in various domains such as healthcare, domotic, intelligent transportation system, agriculture, etc.
S-LOR	Reuses and shares interoperable rules to interpret IoT data.
M3 interoperable domain knowledge	Enriches IoT data with an interoperable knowledge base. It enables to build domain-specific or cross-domain SWoT applications. A subset of this knowledge base is available in M3 templates.
STAC	A security knowledge base and application to identify attacks and suggest security mechanisms related to various technologies (sensor, 3G, web, etc.).

Table 3.1: Description of the M3 framework components

3.2 M3 Architectural Overview

The M3 framework is composed of several layers as following (see Figure 3.3):

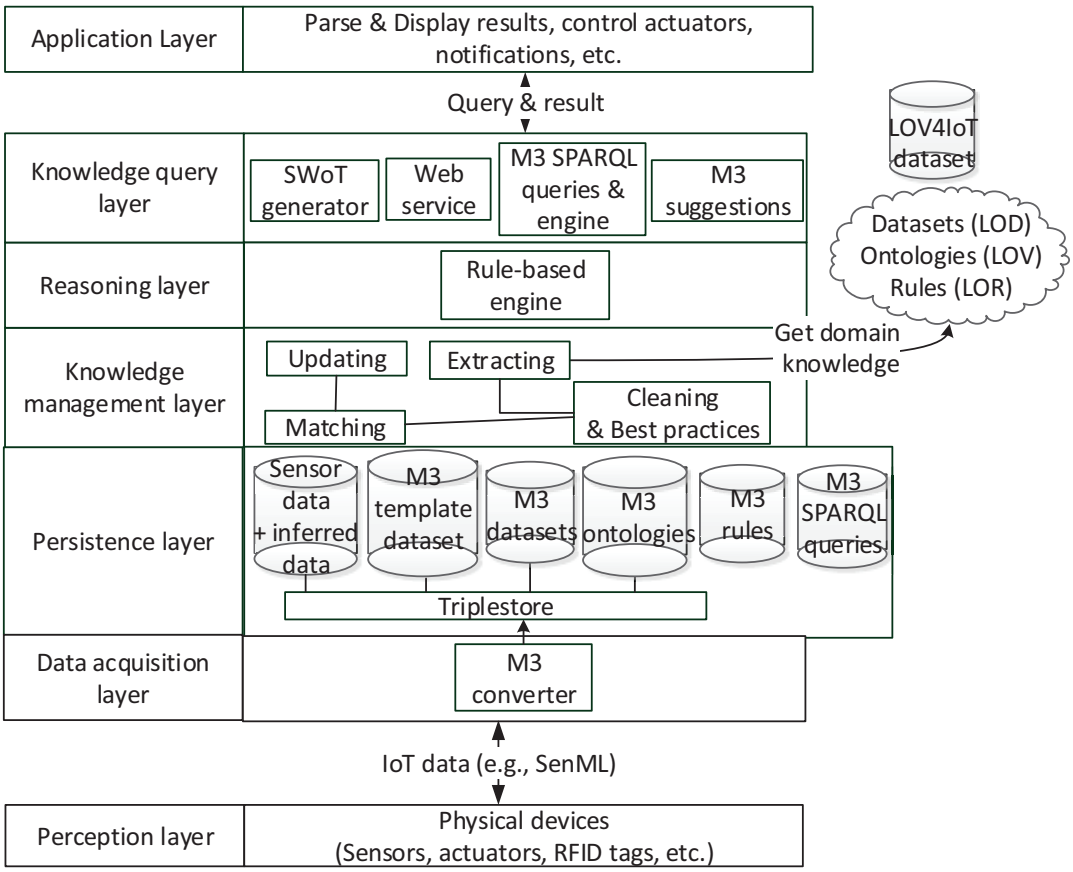

Figure 3.3: Architecture of the M3 framework

- The **perception layer** interacts with physical devices such as sensors, actuators and RFID tags to get their data and control them. In our implementation, this layer is compatible and based on sensor discovery [Datta et al., 2014b] that returns Sensor Markup Language (SenML) data [Jennings et al., 2012]. It could deal with other formats such as Sensor Web Enablement (SWE) [Botts et al., 2008] and could get data generated by other tools such as Graph of Things [Le-Phuoc et al., 2014].
- The **data acquisition layer** gets sensor metadata from M2M devices such as measurement type, unit, sensor type, value and domain. For instance, sensor data is representing in the SenML format: the domain is weather, the measurement type can be cloud cover, the value can be 0 and the unit degree Okta. Okta is the unit to measure the cloud cover. Due to the heterogeneity of measurement descriptions, this

layer converts sensor metadata in a unified description using semantic web technologies such as RDF/XML [Lassila and Swick, 1999]. In this layer, sensor metadata is semantically annotated according to the M3 nomenclature that has been implemented in the M3 ontology. The M3 ontology is an extension of the W3C Semantic Sensor Network (SSN) ontology [Compton et al., 2012], more precisely, an extension of the Observation Value concept. The M3 nomenclature is an essential step to provide a basis for reasoning. This layer produces M3 data thanks to the M3 converter based on the M3 nomenclature.

- The **persistence layer** stores the interoperable M3 domain knowledge (ontologies, datasets and rules), semantic sensor data and inferred sensor data. This layers provides, also, the M3 template dataset to retrieve the M3 domain knowledge to easily build SWoT applications. SPARQL [Prud'Hommeaux et al., 2006] queries have been also designed and are compatible with the M3 domain knowledge to assist developers in querying sensor data. Most of the datasets are stored in a triple store. A triple store is a database for storing semantic data. M3 SPARQL queries and M3 rules are stored in files.

- The **knowledge management layer** is responsible for finding, indexing, designing, reusing and combining domain-specific knowledge (e.g., smart home, intelligent transportation systems, etc.) such as ontologies and datasets to update M3 domain ontologies, M3 datasets and M3 rules which are structured in an interoperable manner. We built a dataset to reference, classify and reuse domain-specific knowledge that we called Linked Open Vocabularies for IoT (LOV4IoT)[1]. LOV4IoT is a knowledge base composed of ontology-based projects relevant for IoT. These projects are based on semantic web technologies and provide domain ontologies, datasets and rules which could be theoretically reused to design domain-specific or cross-domain SWoT applications.

- The **reasoning layer** infers high-level knowledge using reasoning engines and M3 rules stored in the persistence layer. M3 rules are extracted from the LOV4IoT dataset and are redesigned to be interoperable with each other. M3 rules are a set of rules compliant with the M3 ontology to infer new knowledge from M3 data. For instance, when the cloud cover is equal to 0 okta, M3 rules can deduce that the sky is blue. This layer produces smarter M3 data.

- The **knowledge query layer** loads M3 ontologies, M3 datasets, M3 datasets and M3 smart data. Then, it executes SPARQL queries to provide M3 suggestions. SPARQL is a language to query semantic web data. For instance, this layer can suggest activities according to the weather. Indeed, activities are related to weather concepts in tourism and weather datasets.

- The **application layer** employs an application (running on smart devices) which parses and displays the results to end users. For instance, the M3 framework suggests activities according to the weather forecasting (e.g., catamaran when it is windy).

[1] http://www.sensormeasurement.appspot.com/?p=ontologies

Other treatments can be achieved in this layer such as controlling actuators, sending alerts, etc.

3.3 SWoT generator

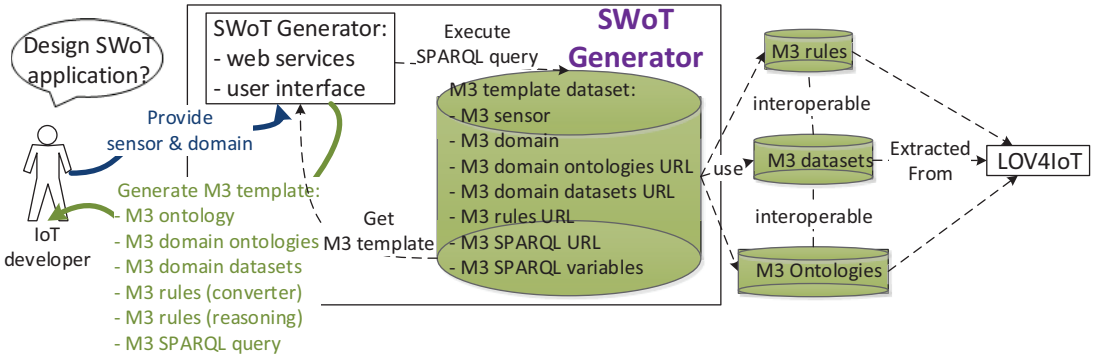

Figure 3.4: Getting M3 templates with the SWoT generator

Our proposed framework is comprised of the SWoT generator to produce templates that will be used to easily design SWoT applications (see Figure 3.4). SWoT is a generator, since it will **generate packages to the developers called M3 templates** with ontologies, datasets, rules and SPARQL queries required to build their IoT applications. The main benefit of the M3 template is to avoid to the developers to learn semantic web technologies, more precisely, they do not need to design their own ontologies and rules or even semantically annotate data. The M3 templates have been manually designed in a dataset to build IoT uses cases. The developers give the name of the sensor used (e.g., LightSensor) and the domain (e.g., Weather) and the SWoT generator looks for M3 templates fitting their needs by executing a SPARQL query on the M3 template dataset. Figure 3.5 shows the SPARQL query to look for templates in the M3 template dataset. The `?m2mdevice` and `?domain` parameters are respectively the sensor used and the domain referenced in the M3 nomenclature and M3 ontology.

The M3 generator automatically produces the M3 template composed of the M3 domain ontologies, M3 datasets, M3 rules and M3 SPARQL queries. Such templates will enable to easily semantically annotate IoT data, interpret it and provide M3 suggestions. The sequence diagram is depicted in Figure 3.6. For the given example (LightSensor and Weather), the M3 framework proposes three cross-domain templates: (1) "Weather, Transport & Safety Devices" suggests safety equipments in the car according to the weather, (2) "Weather, Tourism & Activities" suggests activities according to the weather, and (3) "Weather, Tourism & Clothes" suggests clothes according to the weather. By using the same sensor but in other domain (e.g., Home), the M3 framework proposes 2 other templates: (1) "Home, Presence & Light" to switch on/off the light if someone is detected in

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX m3: <http://sensormeasurement.appspot.com/m3#>

SELECT DISTINCT ?m2mappli ?m2mdevice ?m2mapplilabel ?m2mapplicomment WHERE{
?m2mappli m3:hasM2MDevice ?m2mdevice.
?m2mappli m3:hasContext ?domain.
?m2mappli rdfs:label ?m2mapplilabel.
?m2mappli rdfs:comment ?m2mapplicomment.
FILTER(LANGMATCHES(LANG(?m2mapplilabel), "en"))
FILTER(LANGMATCHES(LANG(?m2mapplicomment), "en"))
}
```

Figure 3.5: The SPARQL query used by the M3 generator to look for M3 templates

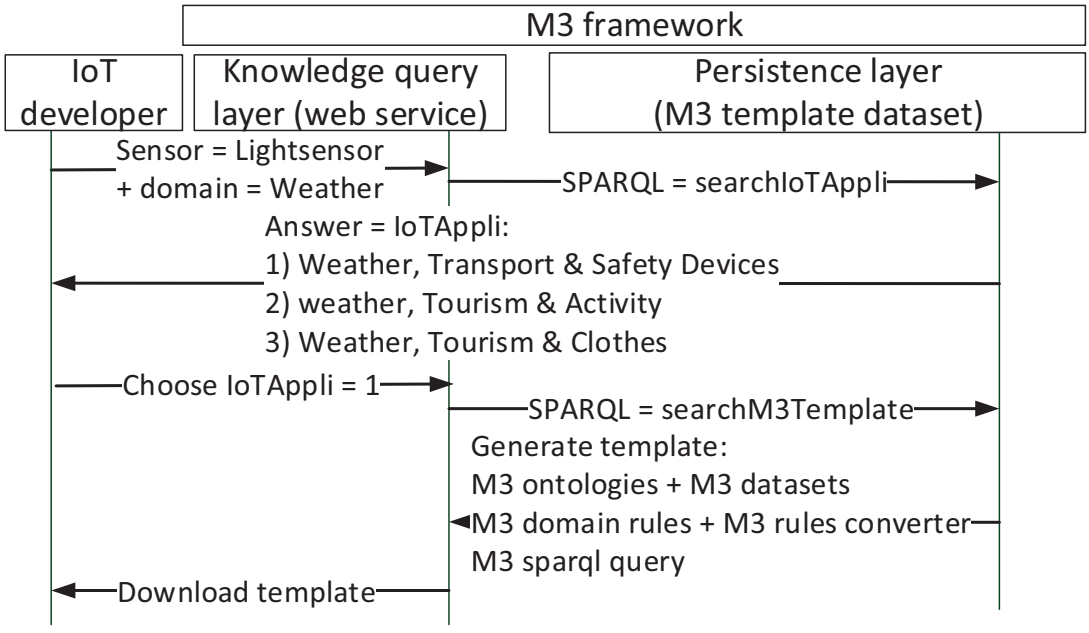

Figure 3.6: Sequence diagram of generating M3 templates

the room , and (2) "Just interpret luminosity values" a domain-specific template to interpret luminosity values. The M3 templates are defined in the M3 template dataset. For each template, we indicate the sensors used, domains, M3 domain ontologies, datasets and rules relevant to build a SWoT application. In Figure 3.7 is displayed an RDF extract of a template. The RDF dataset is available online[2].

```
<m3:M2MApplication rdf:about="&m3;WeatherTransportationSafetyDeviceLight">
   <m3:hasContext rdf:resource="&m3;Weather"/>
   <m3:hasContext rdf:resource="&m3;Transportation"/>
   <rdfs:label xml:lang="en">Luminosity, Transportation and Safety Device</rdfs:label>
   <rdfs:comment xml:lang="en">IoT application to suggest safety devices </rdfs:comment>
   <m3:hasM2MDevice rdf:resource="&m3;LightSensor"/>  => Sensor used
   <m3:hasUrlOntology rdf:resource="&m3;"/>                          => M3 ontology to load
   <m3:hasUrlOntology rdf:resource="&weather;"/>                     => M3 weather ontology to load
   <m3:hasUrlDataset rdf:resource="&weather-dataset;"/>              => M3 weather dataset to load
   <m3:hasUrlOntology rdf:resource="&transport;"/>                   => M3 transport ontology to load
   <m3:hasUrlDataset rdf:resource="&transport-dataset;"/>            => M3 transport dataset to load
   <m3:hasUrlSparql rdf:resource="&sparql;m3SparqlGeneric.sparql"/>  => M3 SPARQL query to execute
   <m3:hasSparqlVariableinferTypeUri rdf:resource="&m3;WeatherLuminosity"/>
   <m3:hasSparqlVariabletypeRecommendedUri rdf:resource="&transport;SafetyDevice"/>
   <m3:hasUrlRule rdf:resource="&lorWeather;"/>  => M3 rules to infer high-level abstractions
   <m3:hasUrlRule rdf:resource="&ruleM3Converter;"/>
</m3:M2MApplication>
```

Figure 3.7: M3 template example implemented in the M3 template dataset

3.4 Designing Interoperable Semantic Web of Things Applications with M3

The M3 framework assists IoT projects, standards and developers in designing interoperable Semantic Web of Things applications as explained in Figure 3.8. Developers offer real and reliable sensor data, represented in SenML format, which can be enriched with our M3 framework. The developers have three main tasks to follow: (1) generate the M3 template, (2) semantically annotate IoT data, and (3) interpret IoT data and get smart M3 suggestions. They can use M3 web services (see M3 API documentation[3]) or even the user interface (see M3 user guide[4]) for these different tasks. **The main added value of the M3 framework is to enable developers designing Semantic Web of Things applications without learning semantic web technologies.** Further, the framework is extensible since we can add more templates and reference more sensors, ontologies and rules to satisfy the needs of the developers.

[2] http://sensormeasurement.appspot.com/dataset/iot-application-template-dataset
[3] http://www.sensormeasurement.appspot.com/documentation/M3APIDocumentation.pdf
[4] http://www.sensormeasurement.appspot.com/documentation/UserGuide.pdf

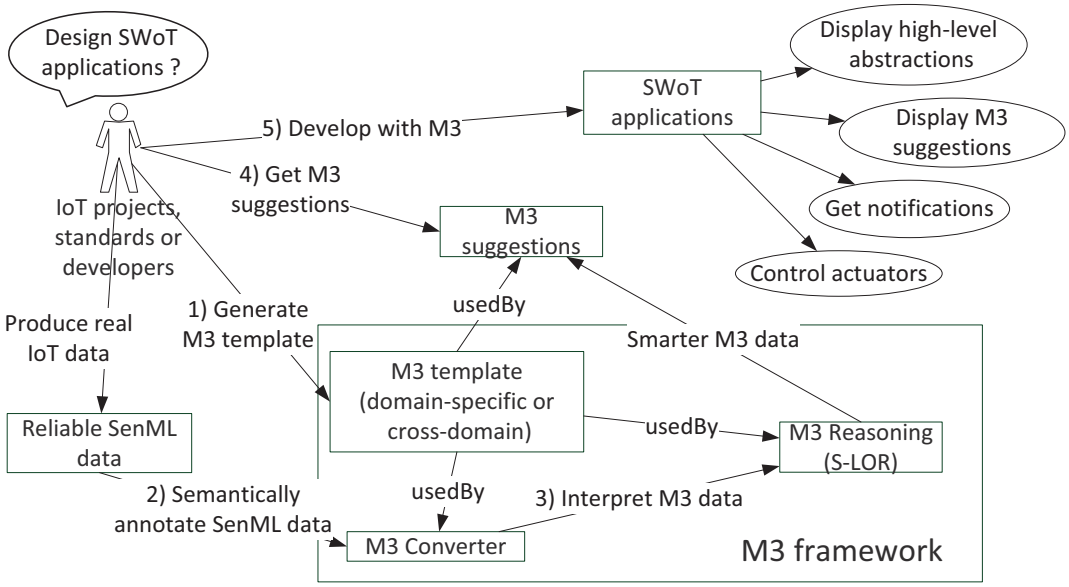

Figure 3.8: Designing SWoT applications with M3

3.4.1 Generating M3 templates

The M3 template fitting developer's needs can be downloaded as a ZIP file through the user interface as displayed in Figure 3.9. The developers choose a sensor and the domain where it is deployed. According to this domain and to the chosen sensor, the SWoT generator provides several M3 templates. The developers choose, then, one template to retrieve from the M3 framework the domain knowledge required to interpret IoT data, enrich and combine it with interoperable domain knowledge. Instead of using the user interface, we provide to the developers an Application Programming Interface (API) with web services to download the template. Currently, we have a dataset composed of 32 M3 templates including 17 cross-domain templates and 15 domain-specific IoT templates.

Figure 3.10 shows the first step to generate the M3 template via web services. Such code will avoid that the non-semantic web expert developers will write any ontologies, rules or datasets. The web service enabling the generation of templates requires two parameters: the M3 sensor and the M3 domain. When giving these two parameters, the developer must be compliant with the M3 nomenclature[5].

3.4.2 Semantically annotate IoT data

The developers can use the M3 converter user interface[6] (see Figure 3.11) or M3 web services to semantically annotate sensor data. The developers give SenML data to the M3 converter

[5] `http://www.sensormeasurement.appspot.com/documentation/NomenclatureSensorData.pdf`
[6] `http://www.sensormeasurement.appspot.com/?p=senml_converter`

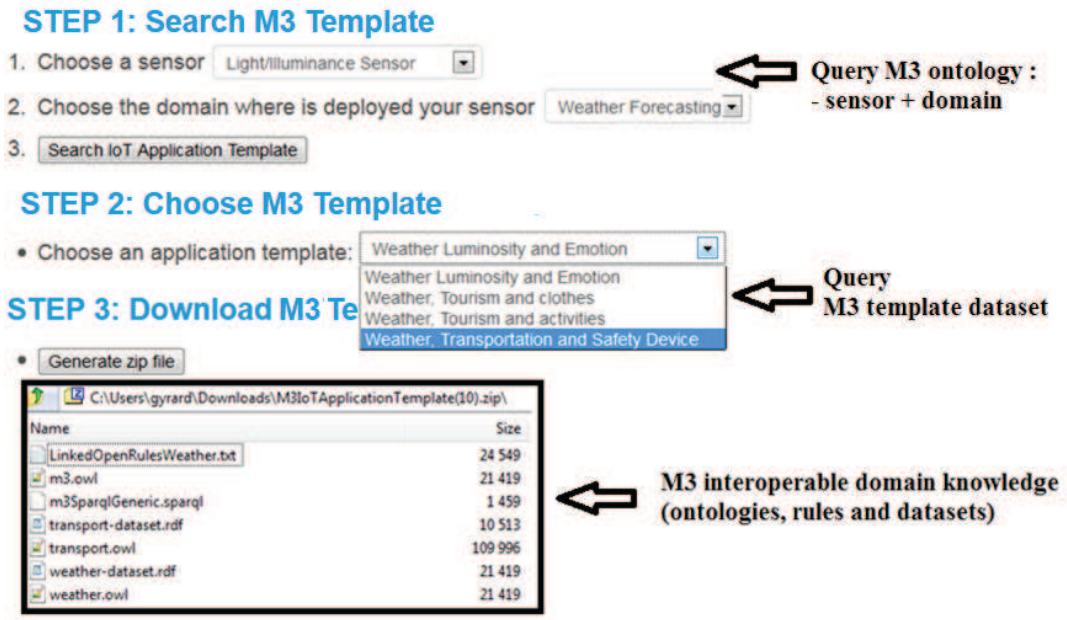

Figure 3.9: Generation of M3 templates with the SWoT generator user interface

to get M3 data that will be compliant with the M3 framework to be easily processed and used by other M3 components.

Figure 3.12 shows the second step to convert IoT data according to the M3 nomenclature. Such code will automatically semantically annotate IoT data with RDF, RDFS and Ontology Web Language (OWL) [Welty et al., 2004] and generate M3 data. The developer does not need to learn such languages. To get better results, SenML data must be compliant with the M3 nomenclature.

3.4.3 Interpreting IoT data

The developers are assisted to design their Semantic Web of Things applications as depicted in Figure 3.13. We guide the developers to use the Jena framework, mainly used to load M3 ontologies, datasets and rules (generated in the M3 template). Then, the developers execute the M3 SPARQL query (generated in the M3 template) to get high-level abstraction from IoT data and M3 suggestions.

The developers are guided through this code. If required, they follows Jena tutorials to load M3 ontologies and datasets, execute SPARQL queries[7] and execute M3 rules[8].

[7] `http://jena.apache.org/tutorials/rdf_api.html`
[8] `http://jena.apache.org/documentation/inference/`

```
String URL_M3_API = "http://www.sensormeasurement.appspot.com/m3/";

// STEP 1: Searching the M3 template fitting your needs
String m3_sensor = "LightSensor";
// paramameter sensorName according to the M3 nomenclature
String m3_domain = "Weather";
// paramameter domain according to the M3 nomenclature
String format = "xml"; // or json
String search_M3_template = queryWebService(URL_M3_API + "searchTemplate/?" +
                           "sensorName=" + m3_sensor +
                           "&domain=" + m3_domain +
                           "&format="+ format);

// STEP 2: Choosing the M3 template
String m3_iotAppli = parse(search_M3_template);
// e.g.: = "WeatherTransportationSafetyDeviceLight";

// STEP 3: Generating the M3 template
String m3_template = queryWebService(URL_M3_API + "generateTemplate/?" +
                                     "iotAppli="+ m3_iotAppli);
// paramameter m3_iotAppli found in STEP 2

// STEP 4: Getting M3 ontologies, datasets and rules
String[] url_file = parse(m3_template);
for each url_file
    String[] url_M3_ontology = download(url_file);
    String[] url_M3_dataset = download(url_file);
    String[] url_M3_rule = download(url_file);

// STEP 5: Getting the SPARQL Query (with variables replaced)
String m3_sparql = queryWebService(URL_M3_API + "getSparqlQuery/? +
                                   "iotAppli="+ m3_iotAppli);
```

Figure 3.10: Pseudo-code to get the M3 template

3.4.4 Making use of M3 templates for IoT EU projects

We provide to the developers more than 30 M3 templates which have been inspired by EU IoT project scenarios such as IoT-i, CityPulse and IoT.est. We built the Table 3.2 to show how the M3 framework can assist IoT projects in building such scenarios. For instance, in the healthcare domain, IoT-i, CityPulse and IoT.est propose scenarios to interpret health measurements such as blood glucose, temperature, heart rate and send alerts if needed. M3 has templates to design such IoT applications. M3 provides the naturopathy scenario to just interpret health measurements or build smarter applications by providing cross-domain suggestions to remedy to the detected symptoms. M3 templates can assist in designing other scenarios such as smart home and transportation. **The entire table classifying all IoT scenarios and how the M3 templates can assist in building such scenarios is available online**[9].

[9] `www.sensormeasurement.appspot.com/?p=m3_scenario`

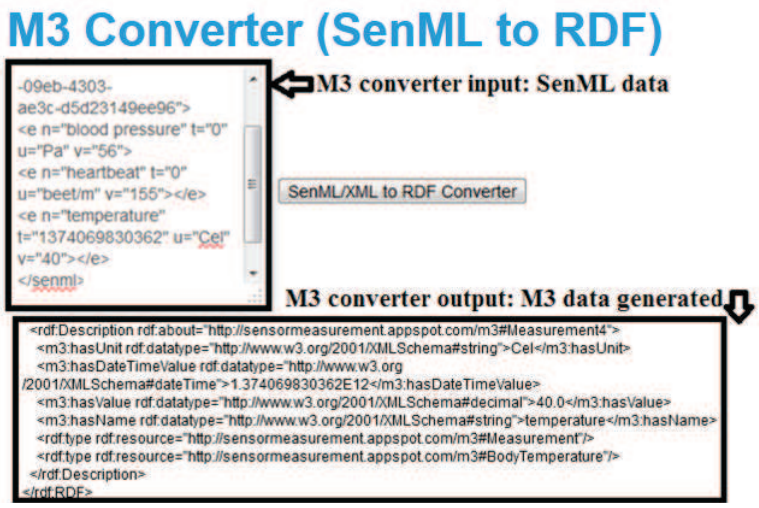

Figure 3.11: M3 converter user interface to generate M3 data

```
1  // Converting your IoT data using SenML to RDF converter
2
3  // String URL_M3_CONVERTER = "http://www.sensormeasurement.appspot.com/swot/";
4  String format = "xml"; // or json
5  String iot_data = getSenMLData();
6
7  String m3_data = queryWebService(URL_M3_CONVERTER + "convert_senml_to_rdf/?data=" + iot_data +
8                                   "&format="+ format);
9
10 store(m3_data);
```

Figure 3.12: Pseudo-code to semantically annotate IoT data

3.5 Integrating M3 in a Semantic-Based M2M Architecture

In this section, we integrate the M3 framework in an architecture inspired by the ETSI M2M architecture. The main goals are to: (1) get sensor measurements from heterogeneous domains, (2) semantically annotate and interpret M2M data, (2) combine domains with each others to build cross-domain M2M applications.

In our proposed semantic-based M2M-based architecture depicted in Figure 3.14, we integrate semantic web technologies both in M2M gateways and M2M applications. We propose two kinds of M2M gateways due to various treatments:

- The **M2M sensor gateways** retrieve M2M measurements provided by heterogeneous M2M devices and include the acquisition interface to support heterogeneous protocols such as RFID, Bluetooth, 6LowPan, CoAP and Zigbee. Several formats can be used such as SenML or SWE to get sensor metadata. We use the lightweight SenML protocol to retrieve heterogeneous sensor measurements for a first and quick implementation. SenML provides simple sensor measurements: the name, the value,

```
// STEP 1: Loading M3 domain knowledge and m3_data
Model model = ModelFactory.createDefaultModel();
InputStream in = new FileInputStream(PATH_FILE + m3_data);
// m3_data has been generated with the M3 converter
model.read( in, fileURL );//read all ontologies generated in the M3 template (.owl)
model.read( in, fileURL );//read all datasets generated in the M3 template (.rdf)
in.close();

// STEP 2: Interpreting M3 data
Reasoner reasoner = new GenericRuleReasoner(Rule.rulesFromURL(PATH_FILE + LinkedOpenRules*.txt));
// LinkedOpenRules*.txt: rules generated in the M3 template
reasoner.setDerivationLogging(true);
InfModel infModel = ModelFactory.createInfModel(reasoner, model); //apply the reasoner
// infModel has been updated with high-level abstraction

// STEP 3: Getting M3 suggestions
//  Executing the SPARQL query:
Query query = QueryFactory(m3_sparql); // m3_sparql has been generated in the M3 template
ResultSet results = QueryExecutionFactory.create(m3_sparql, model)
String m3_suggestions = ResultSetFormatter.asXMLString(results)

// STEP 4: Parsing and displaying m3_suggestions to build the IoT application
// or control actuators, alerting, etc.
```

Figure 3.13: Pseudo-code to interpret IoT data and get M3 suggestions

Project Name	Name application & Goal	Domain & Sensor used	Download M3 IoT application template	M3 Scenario	Reusing domain knowledge (see LOV4IoT web page)
ity Pulse	Chronic disease: Monitor food consumption disease = hyperglycemia	Health domain Body temperature, pulse, blood sugar sensor, RFID on food	M3 IoT Appli: BodyTemperature -> Symptom -> Home remedies (6 rules referenced, naturopathy + health ontology and datasets used)	See Naturopathy scenario	See LOV4IoT section Healthcare ontologies

Table 3.2: M3 templates re-usable for IoT EU project scenarios

the unit and the date (e.g., temperature 5 DegC). SenML or SWE bridges the gap of interoperability of heterogeneous sensor data but does not provide descriptions such as 'this temperature is a body temperature' or 'the milk is produced by cows and contains lactose'. For these reasons, we propose to enrich M2M data with semantic web technologies. The sensor gateways forward the SenML data to the aggregation gateways.

- The **M2M aggregation gateways** semantically annotate sensor metadata with the M3 converter based on semantic web languages (RDF, RDFS, OWL). The M2M aggregation gateway semantically annotate SenML data to provide unified sensor measurements and add an explicit context since data are provided by heterogeneous domains and projects. This step is essential to later interpret data.

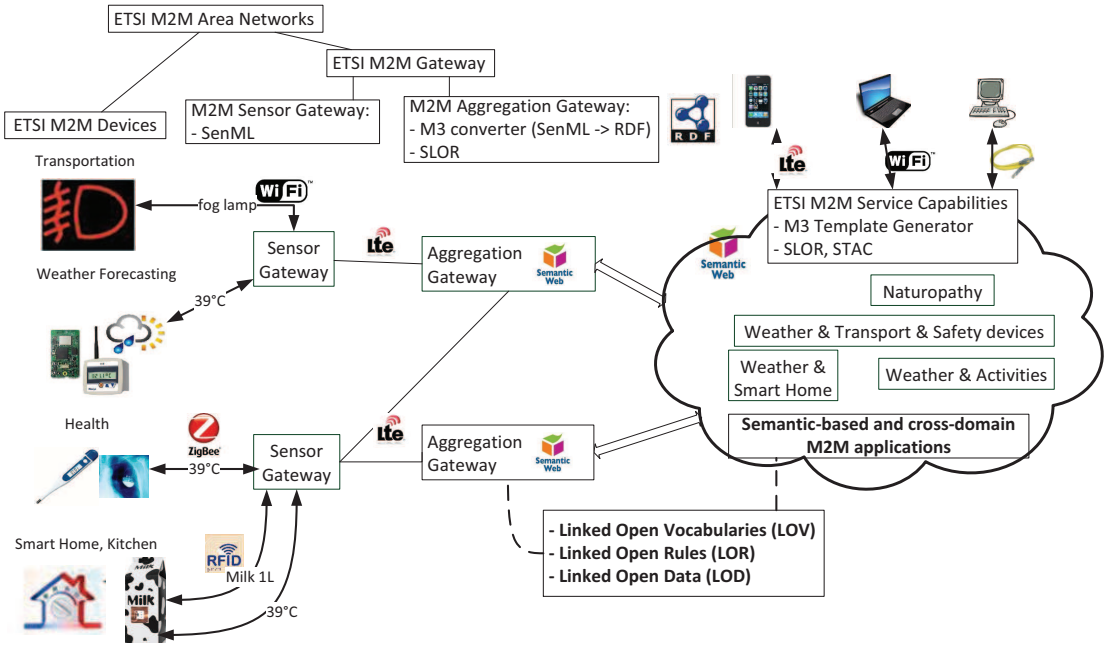

Figure 3.14: Our proposed semantic-based ETSI M2M architecture

Sophisticated semantic treatments are performed in **M2M applications** through semantic-based reasoning with sensor-based Linked Open Rules (S-LOR) and semantic web technologies such as the SPARQL language to query sensor data, the Linked Open Data, the Linked Open Vocabularies and the Linked Open Rules to enrich sensor metadata with external domain knowledge. S-LOR is presented in detail in Chapter 4. M2M applications performed the reasoning on heterogeneous semantic measurements. An example is the naturopathy application to suggest recipes according to the mood, diets, diseases, ingredients available in the kitchen, according to the season, etc. This example shows that four sensor networks need to be merged: health, smart kitchen, weather forecasting and emotion sensor networks.

In the same way, we could integrate the M3 framework in other architectures such as oneM2M, etc.

3.6 Implementation

To demonstrate the feasibility of our approach, we have developed a proof of concept for the M3 conceptual framework which is available online at `http://www.sensormeasurement.appspot.com/`. Figure 3.15 shows the home page of our web site introducing the main M3 components: SWoT generator, LOV4IoT, S-LOR, M3 domain knowledge and STAC. To build semantic web applications, we employed the Jena 2.11 framework [McBride, 2002] which includes the Jena reasoning engine to interpret IoT data and Jena/ARQ to execute SPARQL [Prud'Hommeaux et al., 2006] queries on the M3 domain knowledge base.

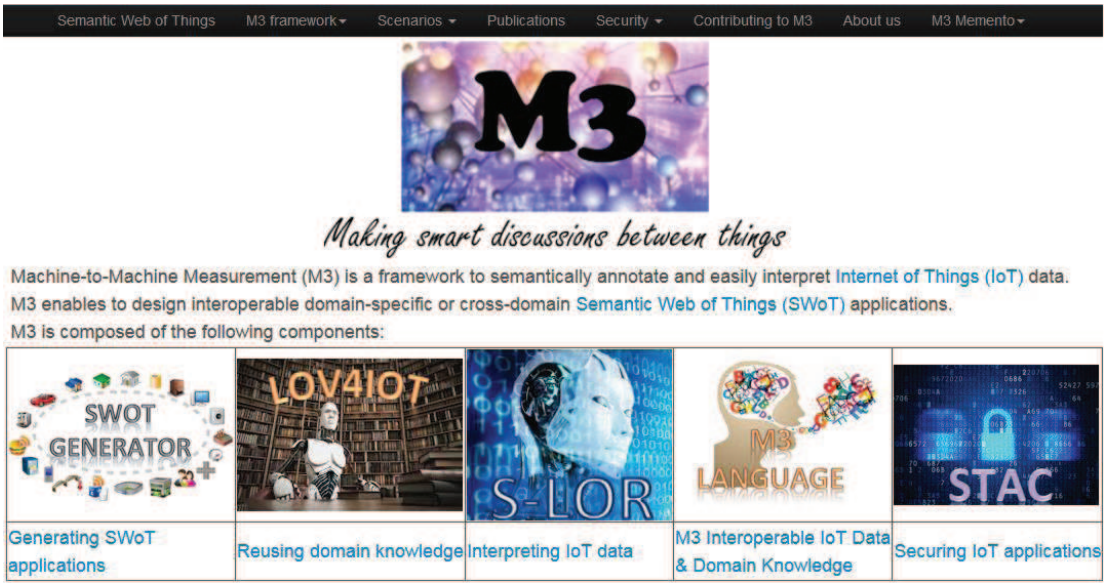

Figure 3.15: Homepage of our proof-of-concept web site

Jena was the easier Semantic Web framework to learn, it was well-documented and had tutorials. Integrating Jena to our application was simple thanks to the JAR file. A light version of Jena is available for constrained devices too. The M3 ontologies and datasets have been developed with RDF, RDFS and OWL. The entire M3 framework has been developed with Java 1.7 and provides RESTful web services thanks to the Jersey implementation. We integrate Google Web Toolkit (GWT) and Google App Engine (GAE) to develop the M3 framework and host the entire web site online. We do not have to maintain a server, it is easy to deploy and maintain applications online. The user interface is implemented with HTML5, CSS3, JavaScript and AJAX technologies to query M3 web services. Finally, the designing phase has been achieved by using extreme programming [Maurer and Martel, 2002] [Lindstrom and Jeffries, 2004] and Scrum-like methodologies.

The M3 framework has been motivated by three cross-domain scenarios: (1) transportation & weather, (2) tourism & weather, and (3) naturopathy which have been inspired by EU projects scenarios such as CityPulse's scenarios[10]. Our cross-domain scenarios demonstrate the importance to combine heterogeneous domains with each other. The technologies used for the implementation of the M3 framework are shown in Figure 3.16.

3.6.1 Scenario 1: Suggesting safety devices according to the weather

As depicted in Figure 3.17, the M3 framework gets basic SenML measurements such as 'the precipitation is 1 millimeter per hour'. Thanks to the M3 converter and S-LOR, M3 infers high-level abstractions from SenML measurements. For instance, Sensor-based Linked

[10] `http://www.ict-citypulse.eu/scenarios/scenarios`

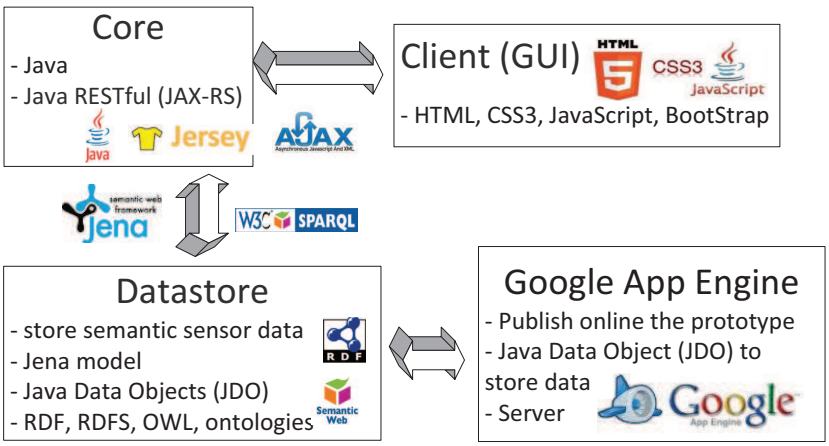

Figure 3.16: Technologies used to develop the M3 framework

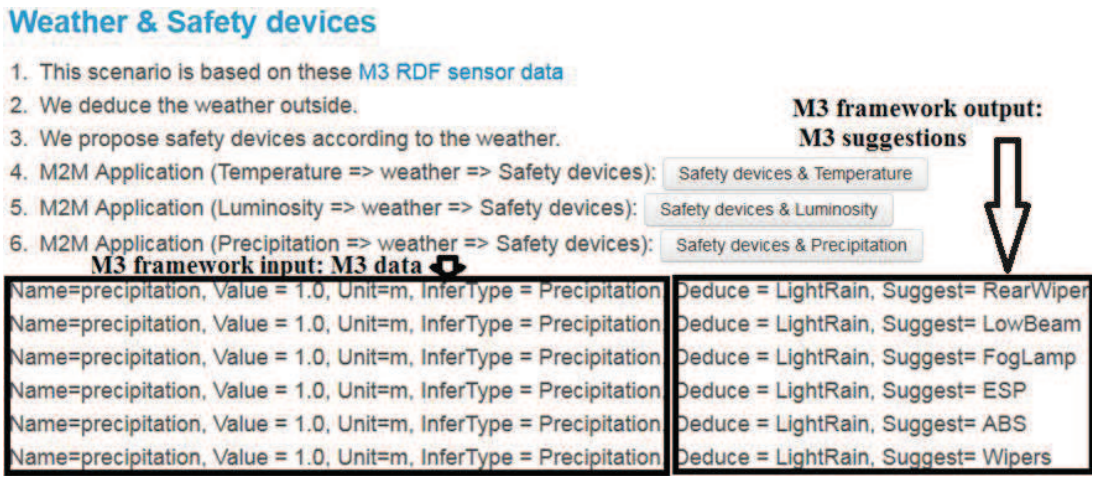

Figure 3.17: M3 suggestions combining transportation and weather domains to suggest safety devices according to the weather

Open Rules (S-LOR) provides six rules to deduce different kinds of precipitation such as no precipitation, extremely heavy rain, light rain, medium rain, heavy rain and tropical storm rain. According to the SenML value, SLOR deduces a light rain. This domain knowledge has been extracted from domain experts such as [Staroch, 2013]. Then, M3 combines this domain knowledge with transportation domain knowledge extracted from [Ruta et al., 2010] [Fuchs et al., 2008b] [Fuchs et al., 2008a] to suggest safety devices to employ in case of rain.

3.6.2 Scenario 2: Suggesting activities or clothes according to the weather

A second cross-domain application use case takes advantage of the ontologies for weather and tourism to suggest activities according to the weather. The M3 framework reused works done by [Reinisch et al., 2011] regarding the weather domain and Chien et al. [Chien et al., 2013] regarding tourism. These works design Semantic Web Rule Language (SWRL)[11] rules that we have redesigned as M3 rules. For instance, if the cloud cover is equal to 0 okta, M3 infers sunny weather [Reinisch et al., 2011] as an high-level abstraction. Then, M3 suggests water activities [Chien et al., 2013] because of the sunny weather.

3.6.3 Scenario 3: Suggesting home remedies according to health measurements

The naturopathy scenario combines healthcare with affective science, food and weather. M3 provides several cross-domain suggestions such as food according to the weather outside or even health measurements. For instance, with a body temperature, M3 deduces a fever and suggests home remedies (e.g., honey) as depicted in Figure 3.18. In this scenario, the

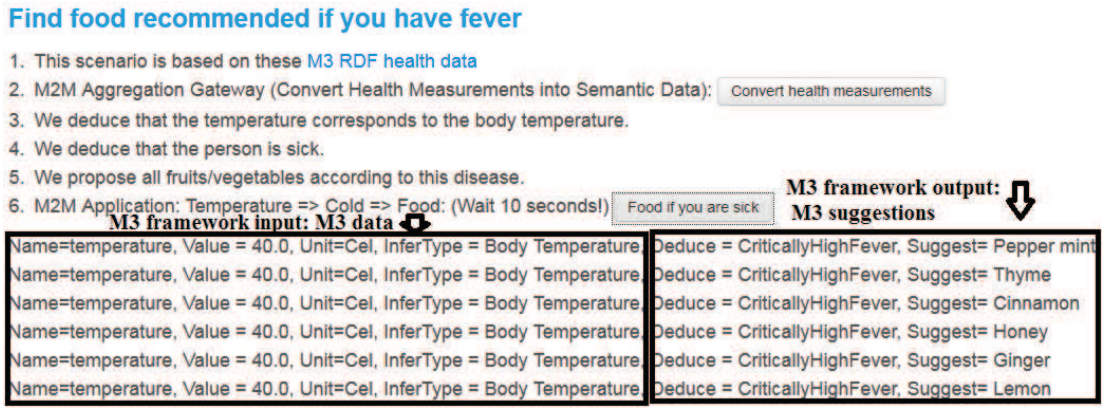

Figure 3.18: The naturopathy scenario suggesting home remedies when a fever is deduced

domain knowledge is extracted from health knowledge expertise [Sharma et al., 2012].

3.7 Evaluation

In this section, we evaluate the M3 framework: (1) by measuring software performances to validate 'Hypothesis 1: The semantic engine is not too resource consuming', (2) with different IoT datasets to validate 'Hypothesis 2: The semantic engine is generic enough to support various kind of measurement' and 'Hypothesis 3: The semantic engine enables

[11] http://www.w3.org/Submission/SWRL/

building cross-domain IoT applications', and (3) with users through Google Analytics to validate 'Hypothesis 4: Users are interested to integrate semantic web technologies to Internet of Things'. Finally, we discuss the evaluation results.

3.7.1 Evaluating software performances

In section 1.5, we introduced 'Hypothesis 1: The semantic engine is not too resource consuming'. To evaluate our proposed approach, we measure the time performed by the M3 converter, S-LOR and when loading IoT data and querying M3 data. The main objective of this thesis is to show the importance to combine the domains with each other and interpret IoT data. Our work has not been tested with very large datasets. The evaluation processing has been performed on the following device: Intel (R) Core (TM) i7 CPU, RAM 4GB, 2GHZ. It has been tested with the transport scenario containing 55 weather rules.

The **M3 converter** has been evaluated by varying the size of IoT datasets. According to the graph depicted in Figure 3.19, M3 is not scalable for gigabytes of data, but is fast enough for small quantity of data to interpret. For instance, for 8 KB of data it takes 20

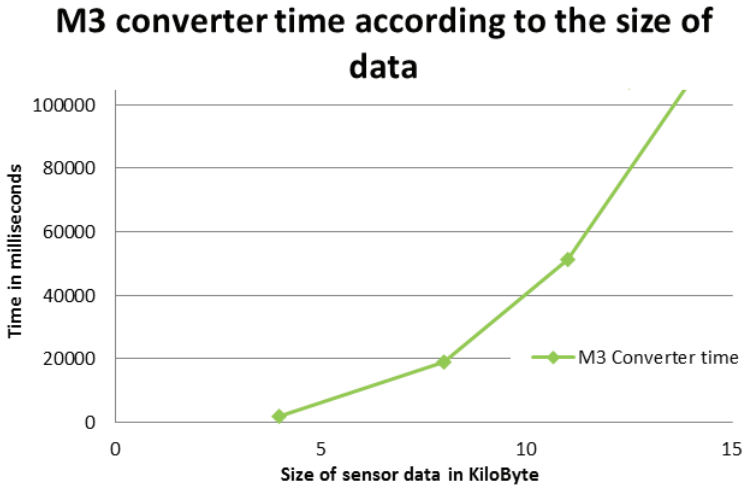

Figure 3.19: M3 converter time according to the size of data

seconds. We used the Java Architecture for XML Binding (JAXB)[12] implementation to convert SenML data to XML and then to RDF. At the beginning of the thesis, we did not use the reasoning engine to convert sensor data according to the M3 ontology. Because of the integration of the Jena reasoning engine to interpret sensor data, we used the same mechanism to annotate sensor data according to the M3 ontology and avoid to have IF THEN ELSE directly in the Java code. Converting sensor data with rules is more flexible for adding new rules.

[12] `https://jaxb.java.net/`

S-LOR has been evaluated by measuring the time needed to interpret M3 data. We changed the number of rules and the size of sensor data. When conducting this evaluation, we had 94 rules. Before the evaluation, we have optimized the number of rule datasets by splitting the 'Linked Open Rules' dataset into sub-datasets classified by domains. Then, we have used the Jena reasoning engine and Jena rules. Our evaluation has been performed with a dataset of 8 KB as it is is shown in Figure 3.20. The performances are good, since the reasoning time takes less than fifty milliseconds even with 50 rules. Another evaluation shows in Figure 3.21 that the performances are still good by varying the size of the sensor datasets, since the reasoning time takes between twenty and thirty milliseconds. Compared to the results from [Barbero et al., 2011], our results are promising. Indeed, they evaluate the reasoning process by measuring performances which is between 0.16 and 0.34 seconds. They have 10 rooms and 101 devices in their Welcoming office scenario.

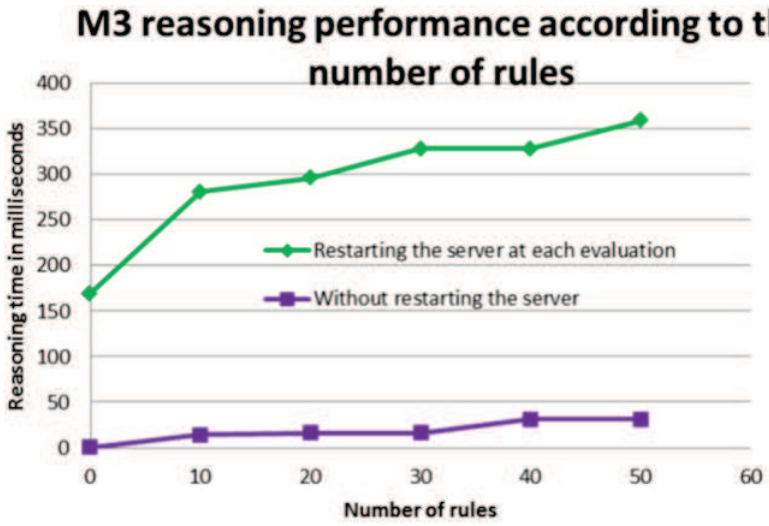

Figure 3.20: M3 reasoning performance according to the number of rules

We measured the time needed to execute SPARQL queries by varying the size of M3 sensor datasets (4, 8, 11 or 14 KB). As depicted in Figure 3.21, the time needed is around 6 or 7 milliseconds to execute the SPARQL query.

To evaluate the **M3 storage**, we stored data in a google database called Java Data Object (JDO) located on the cloud for a rapid prototype. In Figure 3.21 is shown the time needed to retrieve semantic sensor data. It takes between 36 and 37 milliseconds to load different sizes of sensor datasets (4, 8, 11 or 14 KB). We tried to integrate sensor data in a triple store (Jena TDB) but retrieving sensor data is less efficient than with JDO. We have also integrated a local SPARQL endpoint Jena Fuseki to provide access to the M3 domain knowledge and sensor data. Due to incompatibility issues with Google Web application Tookit (GWT), it cannot be published online yet. It has been tested under Apache Tomcat.

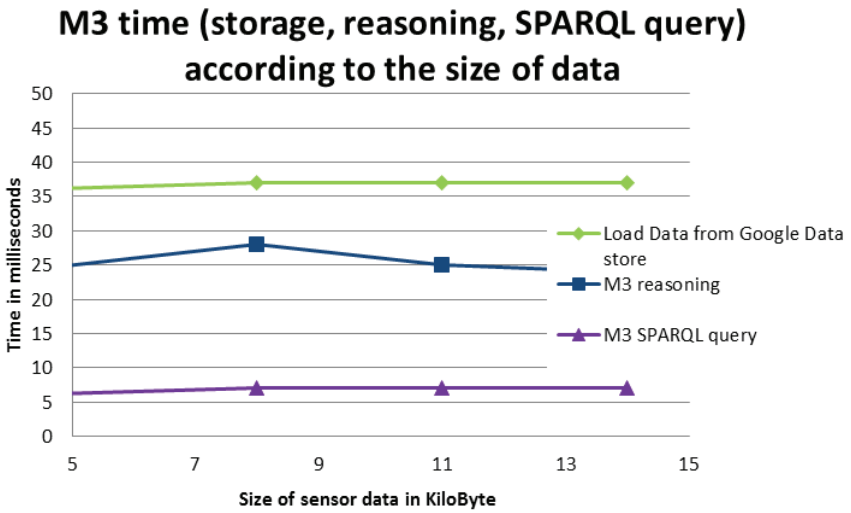

Figure 3.21: M3 tasks time according to the size of data

3.7.2 Evaluating the semantic engine with different IoT datasets

In section 1.5, we introduced 'Hypothesis 2: The semantic engine is generic enough to support various kind of measurement' and 'Hypothesis 3: The semantic engine enables building cross-domain IoT applications'. To evaluate our proposed approach, we evaluate the M3 interoperable domain knowledge provided by the M3 templates. The M3 interoperable domain knowledge has been used to build different domain-specific or cross-domain scenarios based on different sensor datasets (presented in section 3.6). We **have evaluated the M3 framework with 6 datasets** composed of various sensor measurements as depicted in Figure 3.22:

- The weather dataset[13] simulates luminosity, temperature, wind speed, humidity and precipitation measurements. This dataset is used for cross-domain scenarios: (1) smart transportation & weather explained in section 6.3, (2) smart tourism & weather explained in section 6.5, and (3) smart fridge & weather explained in section 6.4.

- The snow dataset[14] simulates only two measurements: precipitation and temperature. This dataset is mainly used to apply more complicated rules which involve two measurements at the same time. This dataset is used for cross-domain scenarios: (1) smart transportation & weather explained in section 6.3, and (2) smart tourism & weather explained in section 6.5.

- The health dataset[15] simulates heart beat, temperature, blood pressure, cholesterol

[13] `http://www.sensormeasurement.appspot.com/dataset/sensor_data/weatherData_8KB_17Septembre2014.rdf`

[14] `http://www.sensormeasurement.appspot.com/dataset/sensor_data/snow_dataset.rdf`

[15] `http://www.sensormeasurement.appspot.com/dataset/sensor_data/senml_m3_health_data.rdf`

and skin conductance measurements. This dataset is used for cross-domain scenarios: smart fridge & weather explained in section 6.4.

- The home dataset[16] simulates sound and temperature measurements. This dataset is used for the home scenario[17].

- The home presence dataset[18] simulates luminosity and presence measurements. This dataset is mainly used to apply more complicated rules which involve two measurements at the same time. This dataset is used for the home scenario[19].

- The location dataset[20] simulates longitude and latitude measurements. This dataset is used for the restaurant scenario[21].

These datasets are semantically annotated witch semantic web technologies and domain knowledge which are provided by the M3 templates. Moreover, rules provided by the M3 templates provide high level abstractions and suggestions. These datasets and the required M3 templates have been used in scenarios that we explain in Chapter 6.

M3 RDF dataset size	Sensor measurements used	Scenario	M3 Suggestions
3 KB	Snow dataset: Precipitation + temperature	Deduce Snow (rule involving 2 sensors), Transport & Tourism	Safety equipment in car, activities and clothes according to the weather
8 KB	Weather dataset: Luminosity, wind speed, temperature, humidity, precipitation	Transport & Tourism	Safety equipment in car, activities, clothes or food according to the weather
3KB	Location dataset: Longitude + Latitude	Restaurant	Find location information & suggest restaurant around
5 KB	Health dataset: Blood pressure, body temperature, cholesterol, heartbeat, skin conductance	Health, Naturopathy	Symptoms, Home Remedies, Diseases
6 KB	Home dataset: Room temperature, sound	Home	Interpret temperature or sound data
3 KB	Home Presence dataset: Luminosity + presence	Home Switch on/off light if nobody (rule involving 2 sensors)	Deduce if someone is in the room or not and switch on/off light

Figure 3.22: M3 framework evaluated with 6 different datasets

[16] http://www.sensormeasurement.appspot.com/dataset/sensor_data/senml_m3_home_data.rdf
[17] http://www.sensormeasurement.appspot.com/?p=home
[18] http://www.sensormeasurement.appspot.com/dataset/sensor_data/presenceLight.rdf
[19] http://www.sensormeasurement.appspot.com/?p=home
[20] http://www.sensormeasurement.appspot.com/dataset/sensor_data/restaurant_lon_lat.rdf
[21] http://www.sensormeasurement.appspot.com/?p=restaurant

3.7.3 Evaluating with end users

In section 1.5, we introduced 'Hypothesis 4: Users are interested to integrate semantic web technologies to Internet of Things'. Even if it is not a scientific evaluation, we have evaluated the M3 framework by sharing on the web our proof-of-concept, our tools and our expertise. According to Google Analytics, we frequently have visitors. This evaluation is important for us, since it shows the effectiveness of this work. According to these results, the work is relevant for other communities and encourage to integrate semantics in Internet of Things. For instance, from December 23th 2014 to January 22th 2015, we had 347 unique page views as depicted in Figure 3.23. The most visited web pages are the home page, followed by LOV4IOT (39 unique page views) and the SWoT generator (22 unique page views). In these results, we have deleted our own consultations: in Google Analytics we added filters to exclude the Antibes city and our Internet Protocol (IP) addresses and even known bots. The average time spent on the LOV4IoT web page is almost 4 minutes and 2 minutes for the SWoT generator.

We have also integrated a visitor map. This map shows that the M3 web site has been visited more than 1545 times from 71 countries since December 5th 2013.

	Page	Pageviews ↓	Unique Pageviews	Avg. Time on Page	Entrances
		476 % of Total: 100.00% (476)	347 % of Total: 100.00% (347)	00:01:37 Avg for View: 00:01:37 (0.00%)	207 % of Total: 100.00% (207)
1.	/ **M3 Main Page**	202 (42.44%)	146 (42.07%)	00:01:23	143 (69.08%)
2.	/?p=ontologies **LOV4IoT**	49 (10.29%)	39 (11.24%)	00:03:49	23 (11.11%)
3.	/index.html **M3 Main Page**	37 (7.77%)	28 (8.07%)	00:00:53	8 (3.86%)
4.	/?p=m3api **M3 Template Generator**	36 (7.56%)	22 (6.34%)	00:01:38	1 (0.48%)
5.	/?p=publication	18 (3.78%)	11 (3.17%)	00:03:03	5 (2.42%)
6.	/?p=architecture	15 (3.15%)	12 (3.46%)	00:01:05	9 (4.35%)
7.	/?p=about_us	13 (2.73%)	12 (3.46%)	00:02:49	1 (0.48%)
8.	/?p=stac	12 (2.52%)	10 (2.88%)	00:00:15	2 (0.97%)
9.	/?p=swot_template **S-LOR**	12 (2.52%)	8 (2.31%)	00:01:37	0 (0.00%)
10.	/?p=transport	12 (2.52%)	8 (2.31%)	00:00:45	3 (1.45%)

Figure 3.23: M3 web site frequently visited

3.7.4 Discussions

The semantic engine comprises the M3 converter, S-LOR reasoning engine and the query engine. The M3 converter is not scalable for gigabytes of data, but implementation optimizations are still possible. Instead of using the JAXB implementation, we could use XSLT to optimize the time needed to semantically annotate M2M data with M3, or use JSON

instead of XML. We consider the performance of the M3 reasoning and execution of the SPARQL queries good. The semantic engine validates 'Hypothesis 1: The semantic engine is not too resource consuming'. The first optimization that has been done before the evaluation was to split the 'M3 rule' dataset into different domains to obtain smaller datasets and optimize performance just by loading the domain needed. Another possible improvement is to select only the rules that we need for specific sensors to reduce the number of rules to apply with the reasoning engine. As a future work, we could automatically extract a lightweight subset of the M3 domain knowledge to reduce the size of M3 templates and M3 processing as much as possible.

Evaluating with different IoT datasets validates 'Hypothesis 2: The semantic engine is generic enough to support various kind of IoT measurement' and 'Hypothesis 3: The semantic engine enables building cross-domain IoT applications'. This evaluation is really encouraging and demonstrates the genericity of our approach to deal with different scenarios. However, such datasets were based on the SenML protocol and the M3 nomenclature. In a future work, we plan to support more heterogeneous formats.

Regarding Google Analytics result, evaluating with end users is really encouraging and validates 'Hypothesis 4: Users are interested to integrate semantic web technologies to Internet of Things'. We could go further in validating the M3 framework, by requiring a logging account to track each user and ask them to evaluate our work. This will be considered as future work. Wa had numerous benefit to evaluate this work with Google Analytics. Indeed, we frequently analyzed the behavior of real-time users which is help us to improve documentations, emphasize much better this work, add new functionalities, etc.

All the evaluation results show that we should continue to work on improving the M3 framework with more templates and on optimizing software performances. The M3 framework is relevant for different users, not only Semantic Web of Things experts, but also developers, IoT experts, semantic web experts, domain experts looking at domain ontologies, etc.

3.8 Concluding Remarks

We described in this chapter the M3 framework to assist IoT developers or even projects to design interoperable Semantic Web of Things applications. Further, we explained that M3 could be easily integrated in an ETSI M2M architecture. Finally, our approach is not just theoretical, but also practical since it has been implemented on the cloud. We will demonstrate in Chapter 6, that this framework has been used by Android developers to design cross-domain mobile SWoT applications. Further, our evaluation encouraged us to improve the M3 framework and continue research in this topic.

As future work, we would like to integrate real data produced by IoT EU projects freshly released[22], and process it within our framework. We also intent to integrate more complicated scenarios involving more sensors and domains by following IoT EU project scenarios. Another important aspect is to deal with real-time processing. Finally, we could find a way to provide communications between two M3 applications to provide smart

[22] `http://iot.ee.surrey.ac.uk:8080/`

discussions between things. Indeed, things will exchange interoperable information with each other thanks to a common language to describe sensor measurements and a common reasoning (explained in the next chapter).

In the next chapter (Chapter 4), we explain in more details the way to interpret IoT data by reusing domain knowledge expertise.

Chapter 4

Sensor-Based Linked Open Rules (S-LOR)

"Everyone knew it was impossible. Then one day someone came who did not know, and he did."

Winston Churchill

"People who are crazy enough to think they can change the world are the ones who do."

Steve Jobs

In this chapter, we assist developers to address "Challenge B: Interpreting IoT data" which is itself composed of two sub-challenges: "Challenge A: Interoperable IoT data" and "Challenge C: Inter-domain interoperability". We assume in this work that the developers want to interpret IoT data. We have designed the Sensor-based Linked Open Rules (S-LOR) approach to easily interpret IoT data and solve the challenges above. Several challenging tasks had to be done to achieve S-LOR: (1) provide a basis for reasoning, (2) reuse domain knowledge expertise to enrich IoT data, and (3) redesign an interoperable domain knowledge.

This chapter is composed as follows. Section 4.1 explains the need to assist developers in interpreting IoT data. Section 4.2 describes the M3 nomenclature and ontology to provide a basis for reasoning. Section 4.3 explains the Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset, a fundamental step for finding, reusing and combining existing knowledge expertise. LOV4IoT synthesizes and classifies more than 270 semantic-based projects relevant for IoT in various areas such as healthcare, building automation, transportation, agriculture, tourism, etc. Section 4.4, demonstrates the most challenging task to examine interoperability issues and redesign an interoperable M3 domain knowledge extracted from LOV4IoT to provide M3 suggestions. Section 4.7 provides the S-LOR proof-of concept. Section 4.8 evaluates S-LOR, LOV4IoT and the M3 domain knowledge. Section 4.9 concludes this chapter and outlines future work.

We assess the following research questions:

- How to find and reuse the domain knowledge expertise to interpret sensor data?
- How to combine domain knowledge (ontologies, datasets, and rules)?
- Which semantic-based reasoning mechanism (rule-based inference, recommender system, machine learning) should we integrate to interpret sensor data?
- How to extract, reuse and combine rules from existing domain knowledge?
- How to design rules in a unified way to easily share and reuse them?

4.1 Assisting IoT developers in Interpreting IoT Data

In Figure 4.1 the developer uses the M3 converter to semantically annotate IoT data. IoT data is compatible with the M3 nomenclature and ontology, an essential step for an easy interpretation of IoT data. Then M3 data is enriched thanks to the M3 interoperable domain knowledge composed of M3 domain ontologies, M3 datasets and M3 rules. Enriched M3 data is queried to provide domain-specific or cross-domains suggestions to the developer. Finally, the developer will display results in a user-friendly interface, send alerts or even order actuators (e.g., open or close a door). We explain in details these components in the next sections.

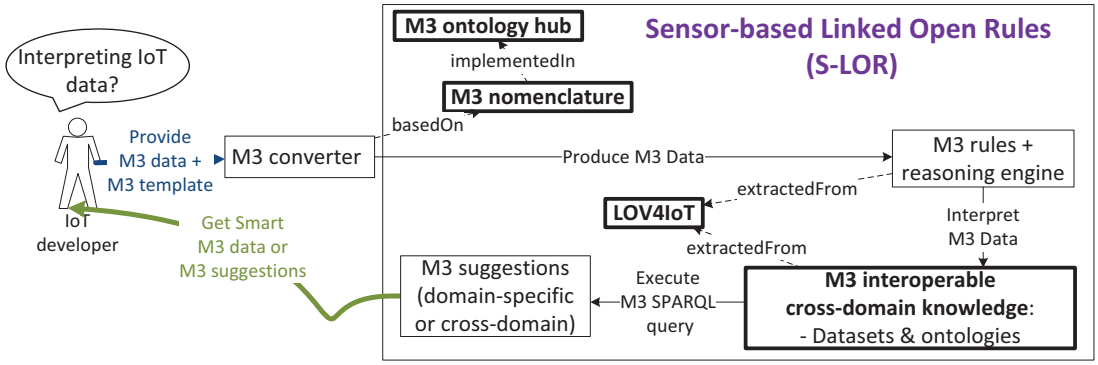

Figure 4.1: Assisting developers in interpreting IoT data with M3

4.2 M3 Nomenclature & Ontology

The M3 nomenclature provides a basis for reasoning. Since sensor data is coming from heterogeneous projects, it is not interoperable. For this reason, we have designed the M3 nomenclature to make the data interoperable and explicitly add the context if needed to

delete any ambiguities. For instance, a temperature could be a body temperature or an external temperature. To achieve this step, we have used semantics, more precisely, we have designed the M3 ontology for the following reasons: (1) to ease interoperability, (2) to add explicit sensor metadata descriptions, (3) to employ reasoning engine to infer new knowledge, (4) to reuse domain knowledge, and (5) to provide a flexible and easy way to update the M3 nomenclature.

The M3 nomenclature has been implemented as an ontology to provide a basis for reasoning and interlinking domains with each other. The M3 ontology synthesizes and unifies all terms to describe sensors, measurements, actuators and domains found in existing projects referenced in LOV4IoT. For instance, precipitation and rainfall sensors represent the same sensor. The uniform descriptions mentioned above are fundamentally necessary to develop cross-domain applications and services. A common nomenclature is described here after and the list is not exhaustive. The second column of the Table 4.1 is the recommended uniform sensor and measurement name, various other names are listed in the third column and units in the fourth column. Table 4.1 presents such a common nomenclature for the sensors used in weather domain. The entire M3 nomenclature is available online[1]. Similar study has been performed for sensors used in health care, smart home, transportation, agriculture, air quality measuring and with actuators. Table 4.2 proposes uniform domain names. The M3 nomenclature has been modeled according to the W3C SSN ontology. More precisely, we provide an extension of `ssn:ObsevationValue`, `ssn:FeatureOfInterest` and `ssn:Sensor` concepts (see Figure 4.2). Further, M3 enables to describe SenML sensor measurements in an interoperable way thanks to this M3 nomenclature (see Figure 4.2). We not only deal with sensors but also RFID tags, actuators, etc. We defined the `m3:Measurement` concept to describe SenML measurement. A measurement has a name (e.g., temperature), a value (e.g., 39) and a unit (e.g., DegreeCelcius). Both the measurement type and the unit follows the M3 nomenclature to provide interoperable sensor data. The uniform M3 descriptions of sensor, measurements and domains have already been communicated to oneM2M WG-5 (MAS)[2] [Gyrard and Bonnet, 2014].

4.3 Linked Open Vocabularies for Internet of Things (LOV4IoT)

The Linked Open Vocabularies for Internet of Things (LOV4IoT) enables reusing domain knowledge expertise. We pursued a deeper analysis of domain knowledge related to sensors and came up with the following research questions:

- Which sensors or actuators are employed?
- What domains do sensors use?
- Which ontologies exist that cover each domain?
- What reasoning exit that cover each domain to interpret sensor data?

[1] http://www.sensormeasurement.appspot.com/documentation/NomenclatureSensorData.pdf

[2] http://onem2m.org/MAShome.cfm

M3 or SenML domain	M3 or SenML sensor/ measurement name	Description, other names (synonyms)	M3 or SenML Unit
Weather	HumiditySensor/ Humidity	Hygrometer, humidity sensor, moisture sensor, soil moisture probes	Percent
Weather	WindDirectionSensor/ WindDirection	Wind direction	DegreeAngle
Weather	SunPositionDirectionSensor/ SunPosition	sun position direction to detect east, west, south, north	DegreeAngle
Weather	AtmosphericPressureSensor/ AtmosphericPressure	Atmospheric pressure sensor, Barometer, barometric pressure sensor	Pascal
Weather	CloudCoverSensor/ CloudCover	Cloud cover sensor	Okta
Weather	SunPositionElevationSensor/ SunElevation	sun position elevation to detect (twilight, day, night, etc.)	DegreeAngle
Weather	SolarRadiationSensor/ SolarRadiation	Solar radiation sensor, par (photo synthetically active radiation) sensor, sun light, solar sensors, sun's radiation intensity	WattPerMeterSquare
Weather	VisibilitySensor/ Visibility	Visibility sensor to detect fog	Miles, Meter
Weather	Thermometer, AirThermometer/ Temperature	Thermometer, temperature sensor, thermistor	DegreeCelsius
Weather	LightSensor/ Luminosity	Light, luminosity, illuminance, lighting	Lux
Weather	PrecipitationSensor/ Precipitation	Precipitation sensor, rainfall sensor, rain fall, pluviometer, rain, rainfall gauge	MilimeterPerHour
Weather	WindSpeedSensor/ WindSpeed	Wind speed sensor, wind velocity sensor, anemometer	MeterPerSecond

Table 4.1: M3 uniform description for sensors in the weather domain

- Is the ontology publicly accessible e.g., downloadable from a website?
- Which technologies or tools are used to implement the ontology or rules?
- Does the ontology follow the semantic web best practices?
- Which projects could be reused and combined to other projects?
- Which security mechanisms are used in the project?

To exploit the domain knowledge expertise and facilitate IoT application development, we have designed the Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset. LOV4IoT references more than 270 ontology-based works related to sensors in various domains such as health care, building automation, food, agriculture, tourism, security, transportation and smart city. We have discovered, identified, studied and referenced these

M3 or SenML Domain name	Description, other names (synonyms)
BuildingAutomation (subclass: Activity)	Smart home, building automation, or building or room (kitchen, bathroom, living room, dining room)
Health	healthcare
Weather	Weather forecasting, meteorology
Agriculture	Agriculture, smart farm, garden
Environment (subclass: Fire)	Environment (earthquake, flooding, forest fire, air pollution)
Emotion	Affective science, emotion, mood, emotional state; brain wave
Transport	Intelligent transportation systems (ITS), smart car/vehicle, transportation
Energy	Smart grid, smart energy
Tourism	Tourism
Location	Location, place, GPS coordinates
City	Smart city, city automation, public lighting
TrackingGood (subclasses: TrackingFood, TrackingCD)	Tracking RIFD goods
Generic	Others

Table 4.2: M3 uniform description for IoT domain names

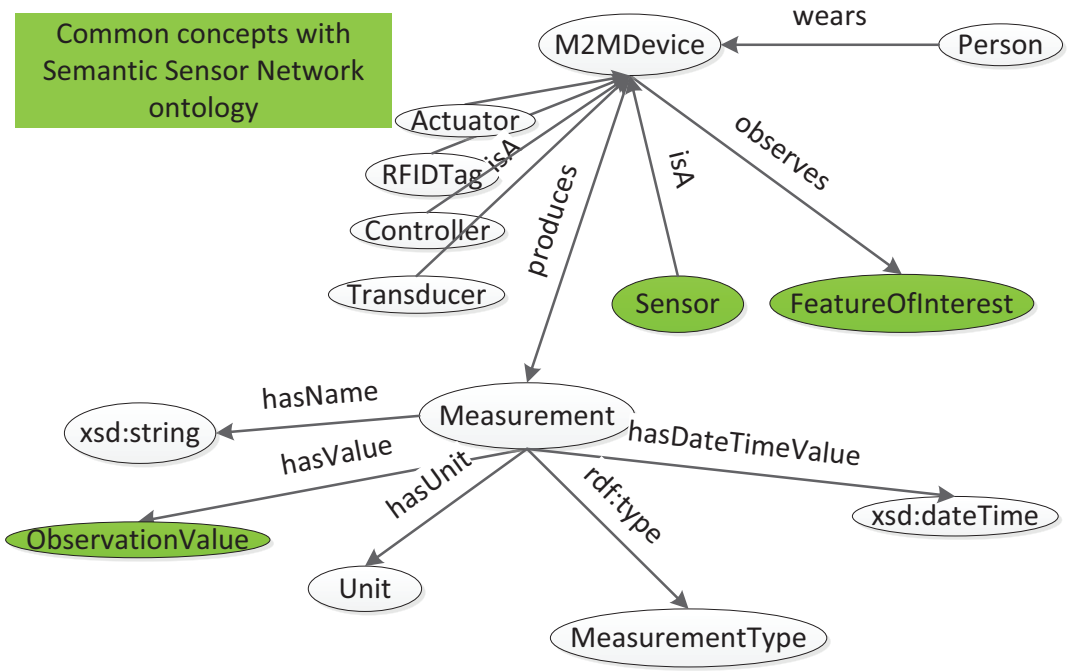

Figure 4.2: M3 ontology, an extension of W3C SSN ontology

works as depicted in Figure 4.3 since: (1) sensors and their measurements are described, (2) they can be used to design new cross-domain use cases (e.g., the naturopathy application to combine health, weather and smart kitchen), (3) the projects are based on ontologies, (4) the projects designed rule-based systems, (5) domain experts published their works in conferences, (6) they explained why they integrate semantics, (7) they describe how they evaluate ontologies, and (8) the ontology or dataset code could be used to implement our scenarios.

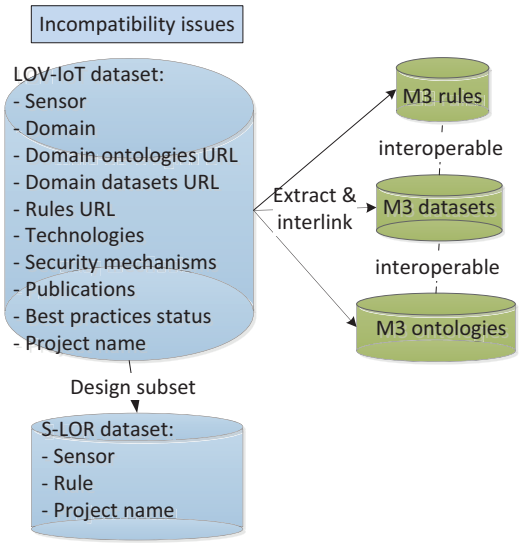

Figure 4.3: The LOV4IoT dataset

Some of these works have been presented in section 2.1. We analyze these works to reuse their ontologies and reasoning mechanisms. Most of the ontologies have been designed with semantic web standard languages such as RDF, RDFS and OWL. Moreover, frequently, the Semantic Web Rule Language (SWRL) has been used for the reasoning.

4.3.1 LOV4IoT, an extension of the LOV catalogue

LOV4IoT is an extension of the LOV catalogue [Vandenbussche et al., 2015], since the ontologies that we classified do not meet the requirements preconized by the LOV catalogue. The ontologies that we referenced in this dataset are not necessarily shared online, but we would like to exploit the knowledge expertise mentioned in the research articles. Requirements preconized by the LOV community such as ontology metadata or adding labels and comments to each concept and property are almost never respected. We contributed to the LOV community, to spread their best practices and encourage the 'sharing and reusing domain knowledge' approach. Unfortunately, we have seen that convincing authors to improve their ontologies is really a time-consuming task.

This limitation could be overcome by improving ontology editors to encourage people

to add labels and comments. Recently, a beta version of ProtegeLOV[3] has been released, an extension of the popular ontology editor which suggests popular ontologies referenced in the LOV catalogue when you are designing a new concept or a new property. The users can directly reuse the concept or integrate `owl:equivalentClass` or `owl:equivalentProperty` links. However, this plugin does not encourage users to add ontology metadata or labels and comments as preconized by the LOV community yet.

For these reasons, we build our own dataset, called Linked Open Vocabularies for Internet of Things (LOV4IoT) to reference and classify ontologies relevant for Internet of Things to: (1) interpret sensor measurements, and (2) combine domains. In our own dataset, we describe the ontology status according to the LOV criteria.

4.3.2 LOV4IoT table

At the beginning of this work, we had few ontology-based projects referenced. We have chosen to classify them in a table and have indicated in each column: the authors, the date of publication, the related research articles, the sensors used, the technologies used, the rules employed and the security mechanisms. Each row in the table has a color to describe the status of the ontologies and rules: (1) lost or confidential, (2) we do not know if we can get the implementations, (3) the authors told us that they will share their knowledge expertise online, (4) the domain knowledge is shared online but do not meet he requirements preconized by the LOV catalogue, (5) the knowledge expertise is shared online and is even referenced on the LOV catalogue since they follow the required best practices. At the beginning, it was just a table in a word document. But, we thought that it can benefit to other people interesting in such ontologies, so we decided to share this classification. The word document has been converted to an HTML web page, by keeping the same idea with the table. It was easier and more flexible to add a new project or move a project to a new domain. Then the number grows more and more. So, we decided to split and classify them by domains.

An extract of the LOV4IoT dataset is available as an HTML web page[4], which is displayed in Figure 4.4. The first column is dedicated to authors and the second to the publication date of the work. In the third column related publications are indicated and in the fourth column the ontology URL if it is provided is given. The fifth column indicates technologies used and the sixth column gives sensor used in the project. Finally, in the seventh column are indicated the rules designed in the projects (e.g., if foggy then safety devices are fog lamp, ESP and ABS). Further, each project is colored according to the ontology status such as the ontology is confidential or lost (in red), try to contact authors to get their ontologies or rules (in white), the authors will soon publish the ontology (in orange), the ontology is online (in yellow), online and referenced in LOV since semantic web best practices are followed (in green). Users such as developers, research engineers or domain experts can surf on this web page to search domain ontologies according to a specific domain.

[3] `http://boris.villazon.terrazas.name/projects/prolov/index.html`

[4] `http://www.sensormeasurement.appspot.com/?p=ontologies`

Authors	Year	Paper	Url onto	Technologies	Sensors	Rules	LOV status	Security
Stocker et al.	2012	Paper: Making sensor of sensor data using ontology: a discussion for road vehicle classification	Trivial ontology: few concepts (response), uses SSN Concepts: Vehicle, Light, Heavy, Dirving Side,	fast fourier transformation, machine learning, Multi Layer Perception (MLP) neural	Vibration, magnetometer, vehicle velocity, camera	rule-based inference (vehicle type)		
Feld, Muller	2011	Paper: The Automotive ontology: Managing knowledge inside the vehicle and sharing it between cars.	Concepts: Road, Parking, Traffic Events, Emotional State, Driving Preferences, Mental State, Abilities,		Speed, voice (microphone), ice sensor, heart beat, blood pressure, arousal, alcohol level			
Hulsen, Zollner, Weiss	2011	Paper: Traffic intersection situation description ontology for advanced driver assistance.	Concepts: Traffic Sign at crossing (right of way sign, yield sign, stop sign), Traffic Light at crossing (green, yellow, red, off), Road connection	RacerPro		Rules: hasOn, CrossingPlain, hasRightOfWay, approached to, connected to, departs from, has entering		
Ruta et al.	2010-2012	Paper: Knowledge-based real-time car monitoring and driving assistance	Ontology and Rules URL Concepts: Weather condictions (fog, windy, cloud, rain, snow, clear), road surface (unever, even), road	owled2 for owl DL ?	GPS, accelerometer, speedometer, wind, esp, abs, fog lamp, rpm sensor, fuel level sensor, maf sensor, throttle load ratio, preoxy sensor, postoxy sensor (oxygen sensor)	OWL restrictions, gas emission & hight density traffic if snow then snow chains, ABS, ESP, low speed if rain then ABS, ESP, low	lov asked 23/03/14, content negociation error (namespace!= uri onto)	

Figure 4.4: An extract of the LOV4IoT dataset displayed in a HTML web page

4.3.3 LOV4IoT RDF dataset

Then, we encountered an issue, sometimes a same ontology-based project could be integrated in both domains such as smart homes and weather (e.g., Staroch et al. [Staroch, 2013]). A table was not enough anymore, we wanted to avoid duplications of the same work in two different domains. Moreover, the number of ontology-based projects attained 200 and the number of domains was growing. We decided to convert this table into a RDF dataset to make statistics on it: the total number of ontologies, the number of ontologies by domains, the number of ontologies according to their status (online, lost, publishing process online, referenced on LOV and contacting authors). Thanks to this RDF dataset, we could also filter ontology-based projects by ontology status or domains, and automatically build a table in the HTML web page, to display a subset of the LOV4IoT dataset according to the user's needs.

The RDF LOV4IoT dataset[5] is available online. An extract of the LOV4IoT dataset in RDF/XML is depicted in Figure 4.5. Users such as developers, research engineers or

[5] `http://www.sensormeasurement.appspot.com/dataset/lov4iot-dataset`

domain experts can make statistics on this dataset or add filter on the dataset. Machines can navigate on the RDF LOV4IoT dataset to easily retrieve the domain knowledge fitting their needs.


```
<m3:M2MApplication rdf:about="PaulStaroch">
   <m3:hasContext rdf:resource="&m3;Weather"/>
   <m3:hasContext rdf:resource="&m3;BuildingAutomation"/>
   <rdfs:label xml:lang="en">[Paul Staroch 2013]. See LOV4IoT for more details.</rdfs:label>
   <rdfs:comment xml:lang="en">Master's Thesis: A weather ontology for predictive control in smart homes. 2013</rdfs:comment>
   <m3:hasM2MDevice rdf:resource="&m3;Thermometer"/>
   <m3:hasM2MDevice rdf:resource="&m3;PrecipitationSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;HumiditySensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;AtmosphericPressureSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;SolarRadiationSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;WindDirectionSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;WindSpeedSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;SunPositionDirectionSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;SunPositionElevationSensor"/>
   <m3:hasM2MDevice rdf:resource="&m3;CloudCoverSensor"/>
   <m3:hasUrlOntology rdf:resource="http://paul.staroch.name/thesis/SmartHomeWeather.owl"/>
   <m3:hasUrlRule rdf:resource="http://paul.staroch.name/thesis/SmartHomeWeather.owl"/>
   <dcterms:creator>
      <foaf:Person rdf:about="mailto:paul@staroch.name">
      <foaf:name>Paul Staroch</foaf:name>
      </foaf:Person>
   </dcterms:creator>
</m3:M2MApplication>
```

Figure 4.5: An extract of the LOV4IoT RDF dataset

Thanks to the RDF dataset, it becomes easy to create SPARQL queries to automatically count the total number of ontologies as depicted in Figure 4.6. We can even count the number of ontologies of each domain and according to the best practices status. Figure 4.6 shows that:

- The total number of ontology-based projects referenced in this dataset is 269.
- The total number of ontology have been classified by domains as follows: 45 for smart home, 8 for smart energy, 10 for activity recognition, 30 for tourism, 28 for transportation, 17 for agriculture, 14 for weather, 3 for smart cities, 17 for sensor networks, 52 for healthcare, 29 for food, 6 for affective sciences, 6 for music, 8 for environments, 7 for fire and 27 for security.
- The total number of ontology have been classified by ontology status as follows: 25 ontologies cannot be shared online (lost, confidential or not implemented), 113 ontologies have been referenced thanks to the research articles that we found but the ontology is not shared online, we are trying to convince authors to share the ontologies or rules on the web, 24 ontologies should be shared online soon according to the authors, 87 ontologies have been published online, most of them because we encouraged authors, 13 ontologies are online and referenced by LOV since the LOV best practices are adopted, and 7 ontologies were already referenced on LOV.

4.3.4 Extracting a dictionary to describe sensor measurements

Based on this tremendous work of classification and synthesization, we built a dictionary to describe popular: (1) sensors or actuators, (2) domains, (3) sensor measurements, and

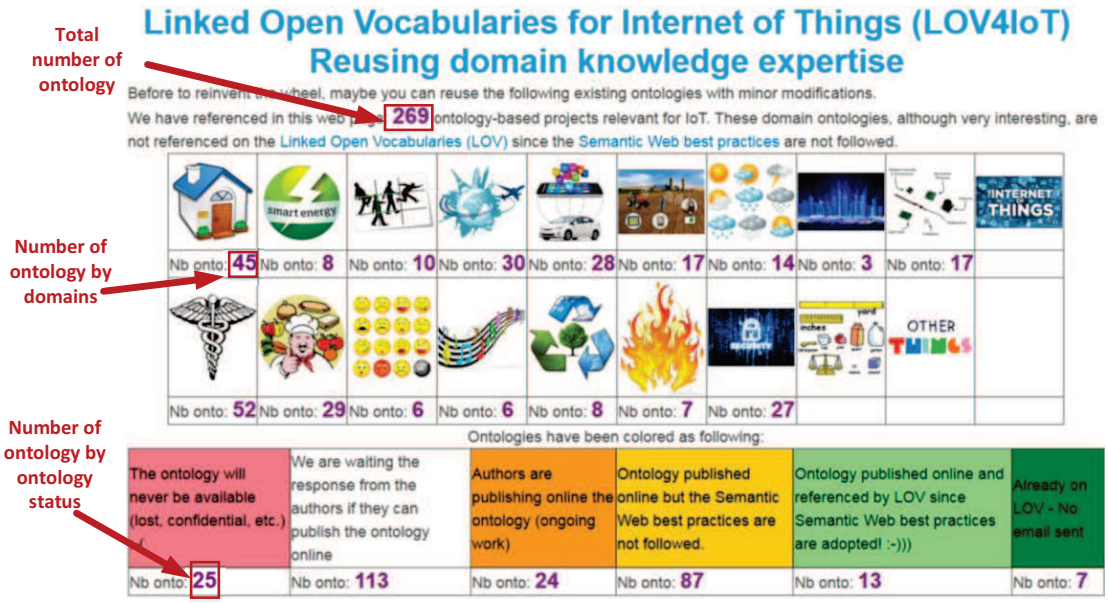

Figure 4.6: Statistics on the LOV4IoT dataset to count the number of ontologies

(4) units. Further, for each sensor measurement, we associated the corresponding units. Further, we frequently found synonyms to describe a same sensor or sensor measurements. We chose the most popular term, and indicate the synonyms in the `rdfs:comment` property in the M3 ontology. The result of this work is the M3 nomenclature and the M3 ontology presented above in 4.2.

4.3.5 Extracting rules to interpret sensor measurements

We have classified and synthesized different languages that have been employed in the ontology-based projects. In Figure 4.7 are displayed heterogeneous softwares and languages:

- Rule Interchange Format (RIF) is proposed a standard format for the 'Linked Rules' [Khandelwal et al., 2011]. RIF is designed by the W3C to unify various rule languages: Semantic Web Rule Language (SWRL), RuleML (Rule Markup Language), R2ML (REWERSE Rule Markup Language) and F-logic [Kifer, 2008]. Seye et al. implement a tool to convert RIF rules into SPARQL CONSTRUCT rules and design a RIF validator [Seye et al., 2012]. However, we did not find any RIF-based implementation tools to extract rules. This language is not popular in the LOVIoT dataset. Only three works mentioned this language.

- The SWRL language [Horrocks et al., 2004] is frequently used by domain experts, since tools have been integrated in the popular Protege ontology editor tool. However, we analyzed the heterogeneity of SWRL rules. Indeed, SWRL rules are not

interoperable because of the heterogeneity among softwares implementing SWRL syntax and reasoning engines. For instance, we referenced 6 protege plugins (SWRL Tab, SWRL DL Safe Rule, SWRLJess Tab, SWRL-IQ, SQWRL and SWRLDroolsTab) implementing SWRL for various reasoning engines.

- The rules can be described as `owl:Restriction` directly in the ontology. This is another kind of implementation of SWRL. Frequently, the rules that we are interested in are implemented in this way.

- SPIN is another language to describe rules, mainly used when the authors used the TopBraid ontology editor. We referenced only 5 ontology-based projects mentioning this language.

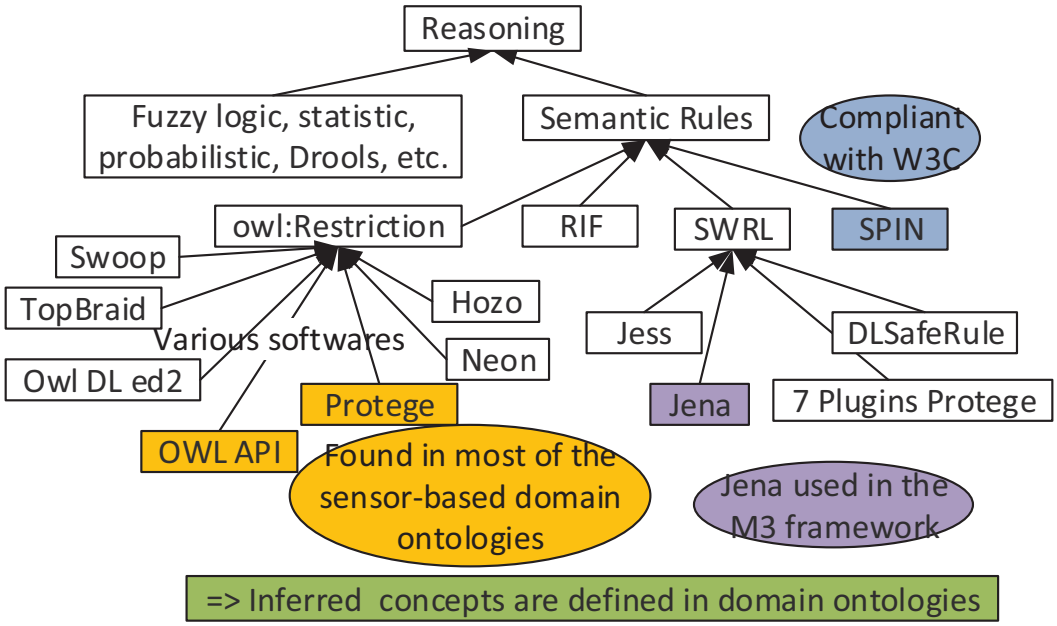

Figure 4.7: Heterogenous rule languages and softwares

Frequently, we found rules described as `owl:Restriction` directly in the ontology (see Figure 4.8). In this figure, the rule described means that if the precipitation measurement is equal to 0 millimeter per hour, then it does not rain. This rule has been described in the weather ontology designed by Staroch et al. [Staroch, 2013]. We expected that all rules are described in this way in all ontologies, but sometimes the unit is not mentioned or the term mentioned to describe the sensor measurement type or the unit does not match to our M3 nomenclature, etc.

This heterogeneity hinders the automatic extraction of rules. For this reason, we manually redesign our own dataset of interoperable rules to infer high-level abstractions from

Figure 4.8: Rule described as an owl:Restriction on ontologies

sensor data. Such rules will be based on the M3 nomenclature and the M3 ontology. Since, most of the existing works design SWRL rules, we use the SWRL language to reuse and combine rules to enrich IoT data. This dataset of rules is called 'Sensor-Based Linked Open Rules (S-LOR)' and is based on the M3 nomenclature and M3 ontology. When the S-LOR dataset attained more than 100 rules, the S-LOR dataset has been split into sub-datasets to classify rules by domains: (1) health, (2) home, (3) weather, and (4) environment. This work is extensible with more rules and more domains. These S-LOR datasets are exploited in the different M3 templates.

4.3.6 Extracting domains

We classified and referenced the most popular domains that we found in ontology-based projects as explained above in Table 4.2 in section 4.2, more precisely: building automation, health, weather, agriculture, environment, emotion, transport, energy, tourism, location, city, tracking good (e.g., tracking food and tracking CD). In the building automation domain, the subclass activity recognition has been defined and in the environment domain, the subclass fire has been added. These domains are described in the M3 nomenclature and ontology and are used in: (1) the iot application template dataset to describe M3 templates, (2) the LOV4IoT dataset to select or count the number of ontologies for each domain, (3) the drop down-list of the SWoT generator when asking to choose a domain, (4) the M3 converter to delete ambiguities and explicitly add the context to IoT measurements, (5) the M3 web services, and (6) in the classification of IoT scenarios explained in section 3.4, sub-section 'Making use of M3 templates for IoT EU projects'.

4.3.7 Lessons learned

At the beginning of this thesis, we thought that it would be easy to reuse and combine these ontologies and rules. Unfortunately, due to heterogeneity, technical issues and limitations of ontology matching tools, reasoning engines and ontology or rule editors, we have not

done the automatic extraction and the automatic linking of this domain knowledge. For these reasons, to show the entire chain to enrich sensor data, we manually re-design our own interoperable knowledge bases to combine rules, ontologies, datasets and domains to infer high-level abstractions from sensor data, as explained in the next section.

Nonetheless, we keep in mind the idea of knowledge extraction for future work. We have analyzed other research fields and tools which can assist us to overcome this challenge. For instance, OWL 2 Rule Template[6] which would enable the detection of rule patterns. We have even found, the DLEJena[7], an implementation compliant with the Jena framework that we could exploit in our M3 framework. Other ideas are to exploit ontology design patterns, ontology methodology, ontology merging tools and ontology alignments.

4.4 Interoperable M3 Cross-Domain Knowledge

M3 interoperable domain knowledge enables extracting and making interoperable the knowledge expertise from LOV4IoT as depicted in Figure 4.9. The first step consists in improving the domain knowledge by following the semantic web best practices. The second step is focused on extracting `owl:Restriction` in domain ontologies and convert them as rules compliant with our M3 framework and M3 ontology. The third and fourth steps are to rewrite the domain ontologies and datasets to be compliant with our M3 framework. The last step consists in integrating ontology matching tools or manually match common concepts to align the domain knowledge to infer additional knowledge and combine domains.

4.4.1 Designing an interoperable M3 domain knowledge

We have redesigned an **interoperable M3 domain knowledge** compatible with our M3 framework (see Figure 4.10) to easily reason on sensor data and build cross-domain IoT applications. To achieve these goals, we had several tasks and semantic web guidelines to follow as explained below. These tasks are executed through the method given in Figure 4.11.

- Unify syntaxes since ontologies, datasets and rules since they are implemented with different ontology editors (Protege, OWL API, Top Braid Composer, OWL DL Ed2, Hozo) and rules editors (7 SWRL Protege Plugins such as SWRLJessTab, SWRL DL-SafeRule, SQWRL, SWRLJessTab) and rule languages (SPIN, RIF, SWRL, SPARQL CONSTRUCT) and rule engine (SWRL Jena rule, Jess).

- Add labels and comments. This is highly recommended for the use of ontology or dataset matching tools. Moreover, it is important to rewrite the domain knowledge in a same language (English) to ease matching tasks and finally add the source, where the domain knowledge comes from to reference the work done by domain experts.

[6] `http://www.w3.org/TR/owl2\discretionary{-}{}{}profiles/#OWL_2_RL`
[7] `http://lpis.csd.auth.gr/systems/DLEJena/`

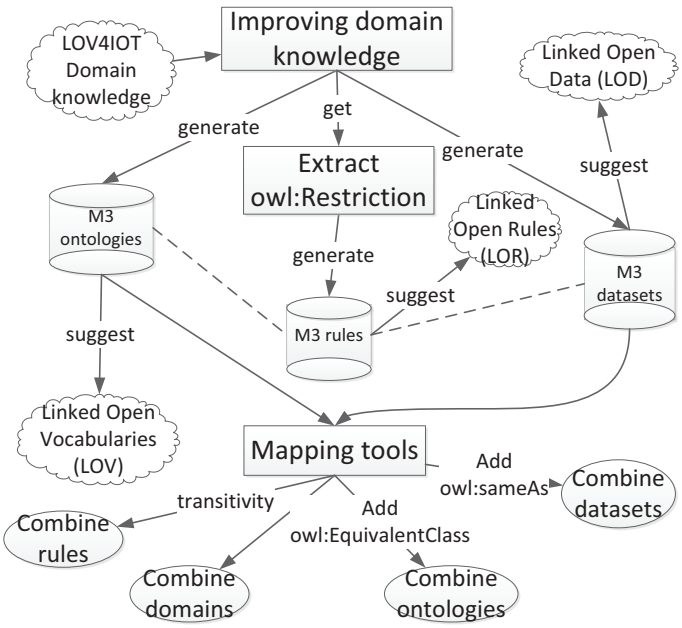

Figure 4.9: Extracting and combining M3 domain knowledge

- Extract rules from domain ontologies. As they are frequently designed as `owl:Restriction`, to convert them according to the Jena syntax, the M3 framework and the M3 ontology.
- Separate ontologies, datasets and rules to ease matching tasks.
- Add links between domain ontologies (e.g., `owl:equivalentClass`) or datasets (e.g., `owl:sameAs`). This step is recommended by the Linked Data best practices [David Wood, 2014] [Heath and Bizer, 2011] and ontology methodologies [Noy et al., 2001].
- Add ontology metadata descriptions proposed by LOV [Vandenbussche and Vatant, 2011] (e.g., title, rights, authors, licenses). To be referenced on LOV, there is a need to improve the domain knowledge with semantic web tools such as the OOPS project[8] [Poveda-Villalón et al., 2012a] [Poveda-Villalón et al., 2012b] to detect common ontology pitfalls. In case of errors encountered when submitting ontologies on LOV, there is a need to check ontologies and fix errors with other tools such as Vapour[9] and TripleChecker[10]. The syntax can also be checked with RDF validator[11] or the Linked Data principles[12] an be exploited to create a well-designed RDF dataset.
- Share the new domain knowledge online and reference it on semantic web tools. M3

[8] `http://oeg-lia3.dia.fi.upm.es/webOOPS/index-content.jsp`
[9] `http://validator.linkeddata.org/vapour`
[10] `http://graphite.ecs.soton.ac.uk/checker/`
[11] `http://www.w3.org/RDF/Validator/`
[12] `http://linkeddata.org/`

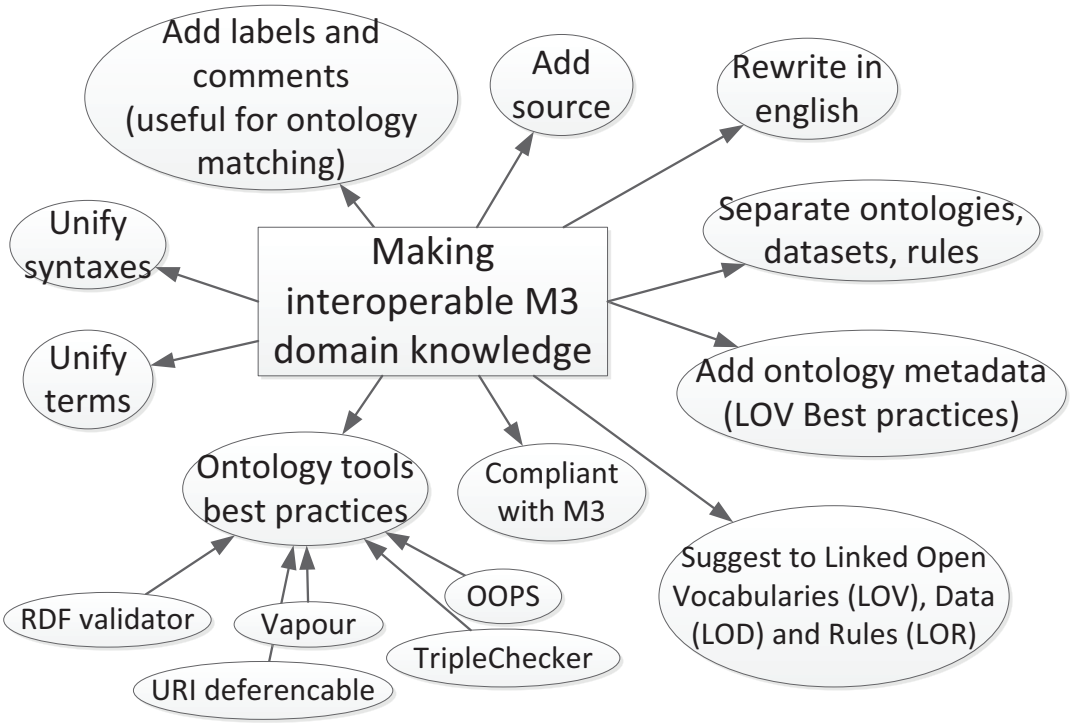

Figure 4.10: Redesigning M3 domain knowledge

rules will be suggested to the Linked Open Rules[13] which is still a work in progress. The M3 domain ontologies could be suggested to the Linked Open Vocabularies catalogue[14] [Vandenbussche and Vatant, 2011] and the semantic search engines such as Watson[15] [d'Aquin and Motta, 2011] and Swoogle. The M3 domain datasets could be suggested to the Linked Open Data, the DataHub project[16] and to semantic search engines such as Sindice[17] [Tummarello et al., 2007]. The name of the ontology (namespace) and the location of the ontology are the same (URI deferencable).

In this thesis, we have retained semantic web best practices and tools such as Methontology [Fernández-López et al., 1997], Ontology Development Guide [Noy et al., 2001], NeOn methodology [Suarez-Figueroa et al., 2012], Common errors & common patterns [Rector et al., 2004] and Linked Data best practices [Heath and Bizer, 2011] [David Wood, 2014] [Janowicz et al., 2014]. These works propose methodologies to build well-structured ontologies or datasets from scratch and suggest to reuse as much as possible existing works by linking them with each

[13] http://www.sensormeasurement.appspot.com/?p=rule
[14] http://lov.okfn.org/dataset/lov/
[15] http://watson.kmi.open.ac.uk/WatsonWUI/
[16] http://datahub.io/fr/
[17] http://sindice.com/

```
Algo Sensor-based Linked Open Rules

Read domain ontology
Unify syntax
Add labels in english (get end URI)
Add comments in english
Add dc:description to reference the source of the original domain ontology

Generate M3 interoperable domain knowledge
    Choose how to represent the domain knwoledge
    Get concepts and properties to generate M3 domain ontologies
    Gets instances to generate M3 domain datasets

Detect and extract owl:Restriction pattern
Convert these patterns into interoperable M3 rules
    Compliant with Jena
    Compliant with M3 ontology
    Compliant with M3 domain ontologies
    Compliant with M3 domain datasets

Add links
    owl:equivalentClass for M3 domain ontologies
    owl:sameAS for M3 domain datasets
    see:Also

Add ontology metadata recommended by LOV
    source, status, author, etc.

Suggest M3 domain knowledge
    Suggest M3 ontologies to the Linked Open Vocabualaries (LOV)
    Suggest M3 datasets to the Linked Open Data (LOD), DataHub
    Suggest M3 rules to the Linked Open Rules (S-LOR)
```

Figure 4.11: S-LOR method

other. We synthesize these methodologies which requires to follow several steps: specification, knowledge acquisition, conceptualization, formalization, implementation, evaluation, documentation and maintenance. These steps are described below.

- **Specification** determines the purpose of the ontology. The goal of our M3 ontology is to interpret interoperable sensor data. The objective of the M3 domain knowledge is to build cross-domain IoT applications.

- **Knowledge acquisition** recommends to reuse existing ontologies. This step has been done by reading more than 270 ontology-based research articles relevant for IoT that have been published in conferences. We have synthesized and classified all these works and detected that they constantly redefine the same domain knowledge.

- **Conceptualization** has been done with an hybrid approach (top-down and bottom up) by defining the most important concepts first and generalize them as much as

possible.

- **Formalization** has been achieved by defining the classes, the class hierarchy and the properties of classes.
- **Implementation** has been accomplished with W3C recommendations such as RDF, RDFS and OWL. It can be achieved with the help of ontology editors such as Protege. We have created M3 domain datasets (instances) according to the M3 domain ontologies.
- **Evaluation** judges the quality of the ontology. It has been explained in details in Section 4.8.
- **Documentation** has been done using labels and comments inside the ontology, even `dc:description` to prove the veracity of the M3 cross-domain suggestions. We documented the M3 domain ontologies using the Parrot[18] tool.
- **Maintenance** of M3 ontologies and datasets has been done continuously through this work.

To design the M3 interoperable domain knowledge, we have followed the process of Suarez-Figueroa et al. as depicted in Figure 4.12 and extend it for each step mentioned above [Suarez-Figueroa et al., 2012].

Figure 4.12: Re-engineering ontologies [Suarez-Figueroa et al., 2012]

Updating the M3 ontology with new sensors, measurements, units or domains is simple and can seamlessly interoperate with the existing environments.

[18] `http://ontorule-project.eu/parrot/parrot`

4.4.2 Combining domain knowledge expertise through M3 rules

One of the main challenging task of the M3 framework is to **combine domain knowledge expertise**. There are three possibilities to combine knowledge: via ontologies, datasets or rules. Frequently, rules could be linked with each other through concepts designed in heterogeneous domain ontologies as depicted in Figure 4.13. Indeed, snow can be described in both weather, smart home or transport ontologies. In the weather ontology, snow will be related to temperature and precipitation, whereas in the transport ontology snow is related to snow chains, low speed, etc. By linking the common concept snow, we link two domains: weather and transportation.

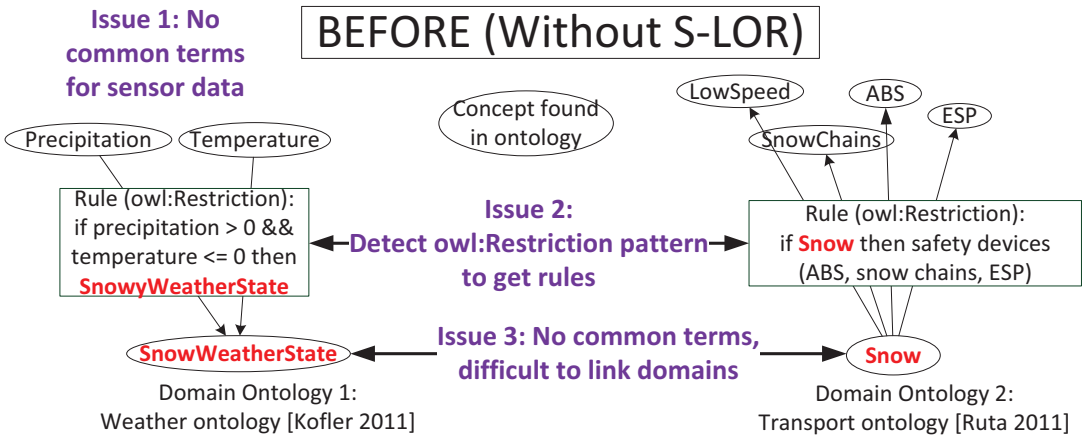

Figure 4.13: Linking rules by linking concepts

However, the main shortcoming is the lack of links between these ontologies. For this reason, we added them in rules.

Figure 4.14 shows the deduction of high-level abstraction from sensor data (e.g., heavy rain). Heavy rain is the result of the reasoning engine by taking into consideration the M3 measurements, M3 units, M3 domains and M3 values. The deduction is also defined in M3 domain datasets (e.g., weather and transportation) which enables to enrich original sensor data. In this example, M3 connects two domains: tourism and transportation thanks to the common terms described in an interoperable manner in M3 rules and M3 datasets.

Figure 4.15 illustrates the grammar of M3 rules, which has been inspired by the Jena rule syntax and structure.

Figure 4.16 shows the process to automatically enrich SenML data to infer high-level abstractions and enrich them with cross-domain datasets. Firstly, we semantically annotate SenML sensor data according to the M3 nomenclature and ontology. Then, interoperable M3 rules are loaded in a reasoning engine with M3 SenML sensor data to infer high-level abstractions. Since common high-level abstractions can be found in heterogenous domains, the domains are easily combined to provide cross-domain suggestions.

In this thesis, we are mainly interested in rule-base reasoning engine (e.g., Jena reasoning engine) to derive high level context information and update the knowledge base with the

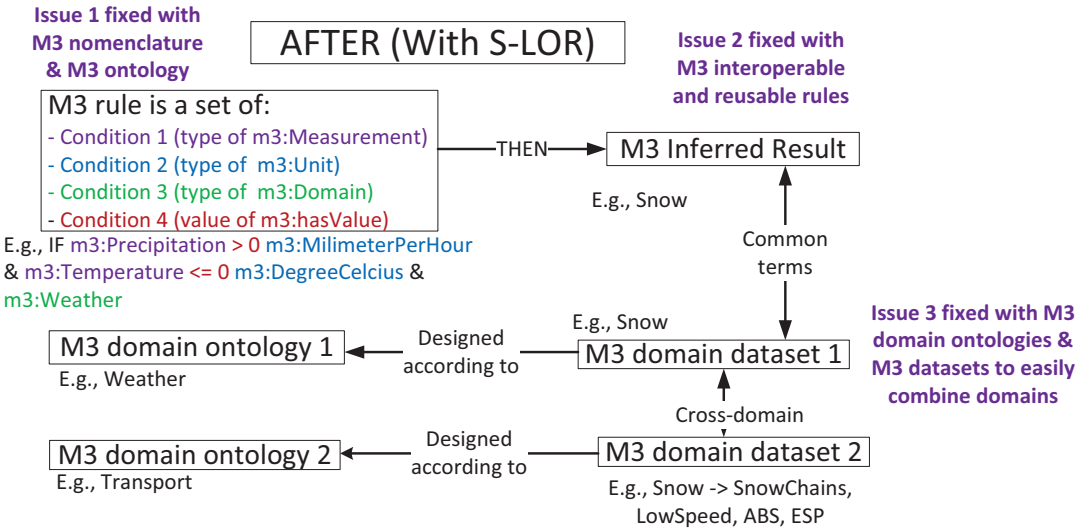

Figure 4.14: Rules for interlinking heterogeneous domain datasets

```
m3-rule := [ ruleName : analyze-m3-data ]

analyze-m3-data := term, ... term -> inferred-data // forward rule

inferred-data := term or [ bare-rule ]

term := (node, node, node) // triple pattern
        or functionComparison(node, ... node) // e.g., greatherThan

node := m3_type
        or uri-ref // e.g. http://foo.com/eg
        or prefix:localname // e.g. rdf:type
        or <uri-ref> // e.g. <myscheme:myuri>
        or ?varname // variable
        or 'a literal' // a plain string literal
        or 'lex'^^typeURI // a typed literal, xsd:* type names supported
        or number // e.g. 42 or 25.5

m3_type := measurement // e.g., m3:BodyTemperature
           or unit // e.g., m3:DegreeCelsius
           or domain // e.g. m3:Health
```

Figure 4.15: Syntax of M3 rules

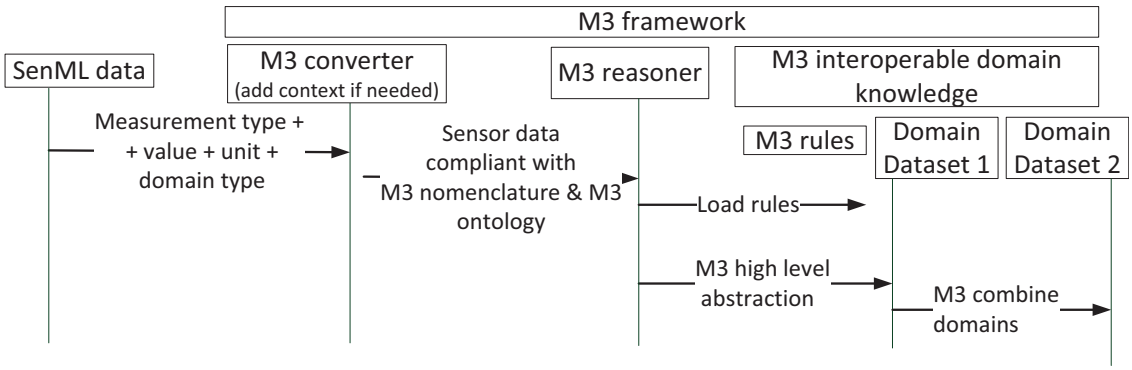

Figure 4.16: Sequence diagram for inferring high-level abstractions with S-LOR

high-level abstractions. Reasoning engines such as Pellet, Racer-Pro, Fact++ have been used to detect and correct inconsistent context information.

Due to ontology matching tools limitations explained in section 2.3.3, we have re-designed the M3 domain knowledge to combine heterogeneous domains with each other when they describe common concepts. Further, the M3 datasets are interoperable with M3 rules and M3 ontologies. This is an essential step to provide M3 cross-domain suggestions. This step is achieved manually to complete the entire M3 process to semantically enrich sensor data and combine domains. As future work, we could improve ontology matching tools, combine them if needed and adapt them to our needs.

4.5 The semantic engine S-LOR integrated in the M3 Approach

Figure 4.17 summarizes the entire process of the Sensor-based Linked Open Rules (S-LOR) approach integrated in the M3 framework [Gyrard et al., 2014b]. In this figure, a same measurement (temperature 38.7 DegC) is described in two different domains: path A for healthcare and path B for weather forecasting. This example highlights the necessity to: (1) explicitly add description to sensor measurements, (2) interpret IoT data, and (3) combine domains to design cross-domain applications. The first box, called 'IoT data' returns sensor descriptions such as temperature 38.7 DegC. Such descriptions are implemented according to the SenML language. Then, in the second box, called 'Semantic IoT data', previous data is semantically annotated according to the M3 nomenclature and ontology, which is required for the future steps. Then, in the fourth box, called 'Semantic Rule, new domain concept', the S-LOR approach is exploited, a set of interoperable rules based on the M3 ontology and nomenclature to infer high-level abstractions. In path A, S-LOR deduces the concept fever, whereas in path B, S-LOR deduces the concept hot. Then, in the boxes 4 and 5, called 'Domain ontologies' and 'Domain datasets' the results of the reasoning provided by S-LOR are linked to the M3 domain ontologies and datasets. Then, in step 6, 'Cross

domain applications', the M3 interoperable domain knowledge is used to combine domains and provides suggestions. For instance, food related to the fever symptom in path A, and food related to season in path B. Since food referred to the same namespace in both domain knowledge, it is easy to combine domains. Finally, in step 7, a SPARQL request queries the M3 interoperable cross-domain knowledge to get smarter data and suggestions.

The provided results will be later parsed and exploited in the final application such as the naturopathy application which suggests home remedies when fever is detected.

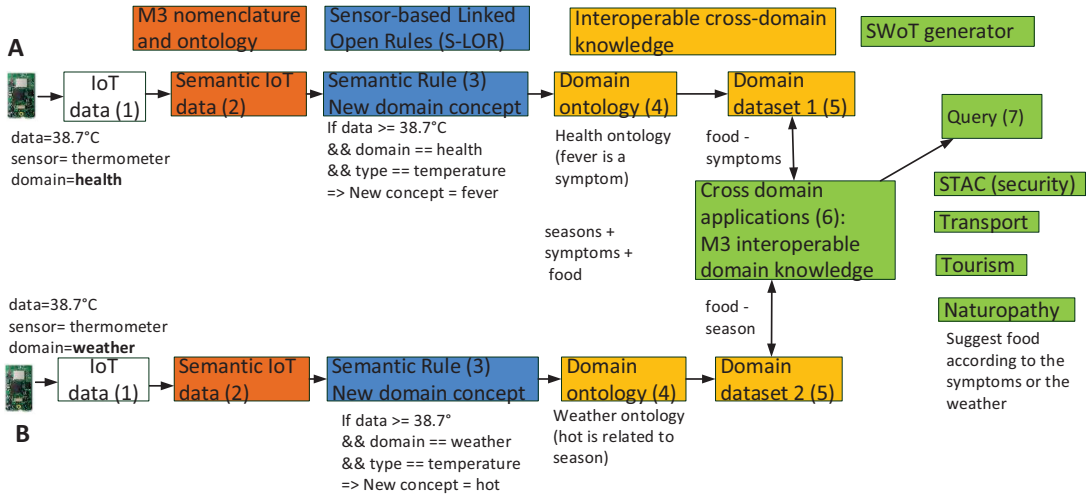

Figure 4.17: S-LOR integrated in the M3 approach

4.6 S-LOR: A 'Share and Reuse' Based Reasoning Approach

Sensor-based Linked Open Rules (S-LOR) is not just a dataset of interoperable M3 rules that is exploited in the M3 templates to build semantic-based IoT applications. This is also an innovative approach, stemming from the 'Linked Open Data' approach, to share and reuse interoperable rules on the Web to interpret IoT data. Currently, data is considered as the new oil. From our point of view, the most important aspect is the high-level abstractions that have been inferred from IoT data. This can be done with rule-based systems, recommender systems or even machine learning algorithms. In this thesis, we are mainly focused on the rule-based systems, since we found rules in ontology-based projects referenced in LOV4IoT. Such datasets could be enriched by domain experts and they could even add ratings on the rules such as trust or popularity. A major challenge is to check completeness (i.e., cover all possible values) and correctness (i.e., no contradiction) of these rules when integrating new rules in the dataset.

As future work, we plan to provide templates to guide users to choose the best systems fitting their needs according to the kind of data that they want to exploit. Indeed, we could guide users to use a specific algorithm to interpret sensor data. For instance, in

[Stocker et al., 2012], the authors use neural networks to classify road vehicles by exploiting measurement of vibration. Wa are aware, that rule-based systems are sufficient for some kind of measurement data. We plan to extend our set of templates with new kind of templates, instead of referencing an URL with a set of rules, we could reference the machine learning required for a specific sensor. For instance, templates related to vibration sensors could reference the neural network algorithm.

Existing approaches such as the KAT toolkit [Ganz et al., 2014] [Ganz, 2014] and Intelligo [Henson, 2013] propose machine learning based approaches to enrich sensor data combined with semantic web technologies. But there is not yet any approaches to share the way to enrich IoT data by suggesting which machine learning algorithm should I use. To solve such challenges, we have in mind to combine KAT and Intelligo to S-LOR to handle the reasoning with more complicated sensors.

4.7 Implementation

A proof-of concept of S-LOR is depicted in Figure 4.18. The drop-down list displays all sensors referenced in the M3 ontology. This is done with a simple SPARQL query, get all `rdfs:subclassOf` of the `m3:Sensor` class. For the implementation, we employ the rule-based Jena inference engine[19] to execute M3 rules to infer high-level abstractions and update the knowledge base. For instance, the user chooses "Precipitation sensor". Then, S-LOR will query its rule dataset to display all rules involving this sensor. S-LOR displays about 20 rules to interpret precipitation values such as `HeavyRain`, `NoPrecipitation`, `MediumRain` or `Snow`. To deduce `HeavyRain`, precipitation values should be between 20 and

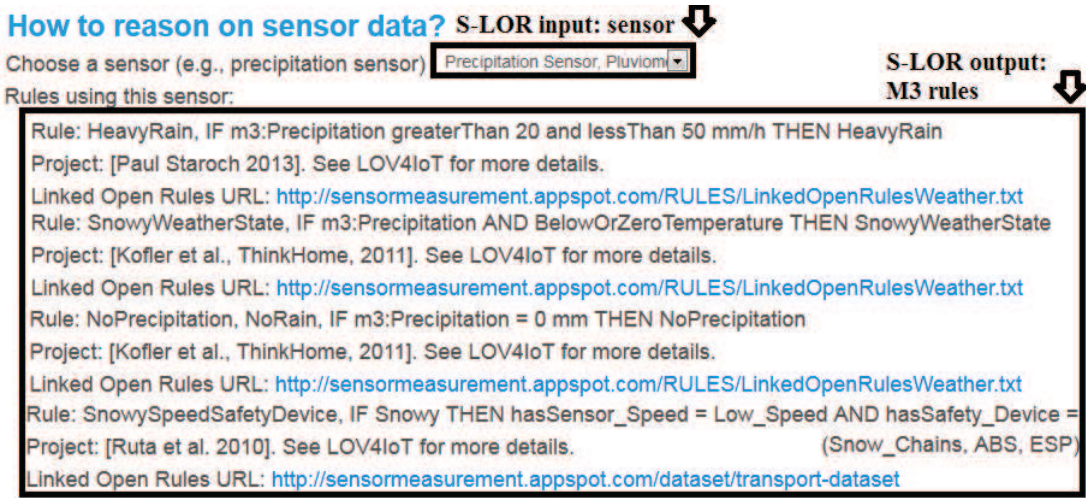

Figure 4.18: S-LOR rules to interpret precipitation measurements

[19] `http://jena.apache.org/documentation/inference/`

50 millimeter per hour. `Snow` is a more complicated rule since it involves two sensors: precipitation and temperature sensors. `Snow` is deduced when precipitation measurements are strictly more than 0 millimeter per hour and temperature measurements less than 0 degree Celsius. These rules are used in different scenarios: suggesting activities, garments or safety devices according to the weather forecasting as explained in section 3.6. A new scenario is depicted in Figure 4.19 to demonstrate that the M3 framework and S-LOR are generic and flexible enough since a reasoning can be performed implying several sensors. This snow scenario suggests activities when it is snowy. It employs S-LOR rules and M3 tourism datasets. The snow rule is extracted from [Staroch, 2013] and the abstraction of activities depending on the weather has been inspired by [Chien et al., 2013], [Choi et al., 2009], [Daramola et al., 2009], [Wang et al., 2008], [García-Crespo et al., 2011] and [Blanco-Fernández et al., 2011].

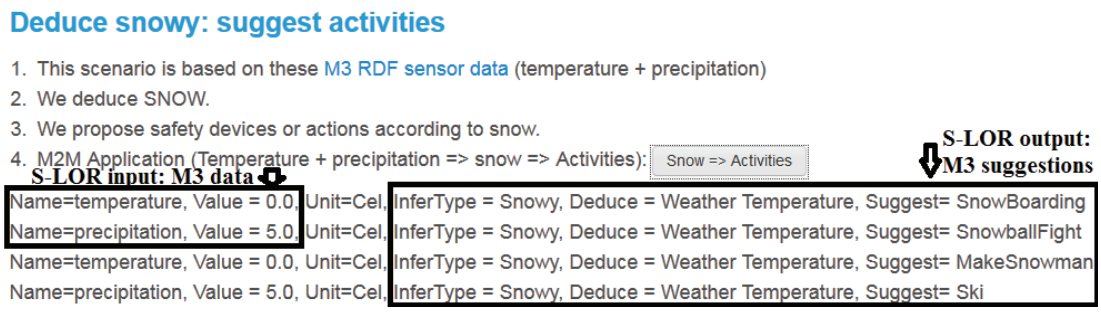

Figure 4.19: Snow & Activity scenario based on the snow rule

For each sensor measurement, we tried to find rules as much as possible. More we have rules, better is the S-LOR dataset. Performance is not an issue in this case, since we can load only the rules that we need and not the entire dataset. It is really fast and simple to update the datasets with new rules. Sometimes, overlapping with other rules appears. In this case, we implement more precise rules. For instance, if a domain expert implements 16 rules to interpret precipitation measurements, and another domain expert only 3 rules, we implement the rules designed by the first expert since they are more precise. A limitation of S-LOR, is that rule-based reasoning is not sufficient for all sensor measurements (e.g., Electrocardiography (ECG), gyroscope). Some sensor measurements require machine learning algorithms to infer high-level abstractions.

4.8 Evaluation

In this section, we evaluate S-LOR by looking at: (1) completeness and correctness of M3 rules to validate 'Hypothesis 5: The dataset of M3 rules is reliable to interpret IoT data', (2) the number of ontologies re-usable from LOV4IOT to validate 'Hypothesis 6: A dataset of ontology-based projects relevant for IoT can be exploited outside of the M3 framework', and (3) the M3 domain knowledge with semantic web methodologies and tools

to validate 'Hypothesis 7: The knowledge base built to interpret IoT data encourages the interoperability of data and domains'.

4.8.1 Evaluating M3 rules with completeness and correctness

In section 1.5, we introduced 'Hypothesis 5: The dataset of M3 rules is reliable to interpret IoT data'. We **evaluate S-LOR** by looking at completeness and correctness of rules. **Correctness** means that there are no incompatibility with other rules. **Completeness** means that all sensor values are covered by a high level information. In table 4.3, a new

M3 or SenML domain	M3 or SenML sensor/ measurement name	Description, other names (synonyms)	M3 or SenML Unit	M3 rules
Weather	HumiditySensor/ Humidity	Hygrometer, humidity sensor, moisture sensor, soil moisture probes	Percent	Correctness OK (Conflict resolved with [Kofler 2011] and [Rodriguez 2014]) + Completude OK (5 rules [Staroch 2013])
Weather	WindDirectionSensor/ WindDirection	Wind direction	DegreeAngle	Completude OK (5 rules [Staroch 2013]) + Correctness OK
Weather	CloudCoverSensor/ CloudCover	Cloud cover sensor	Okta	Completude OK (5 rules [Staroch 2013]) + Correctness OK (even with [Kofler 2011])
Weather	SunPositionElevationSensor/ SunElevation	sun position elevation to detect (twilight, day, night, etc.)	DegreeAngle	8 rules [Staroch 2013] Completude NO + Correctness NO

Table 4.3: Evaluating S-LOR with completeness & correctness

column dedicated to completeness and correctness is added to the M3 nomenclature, which indicates the related rules to each sensor. For instance, M3 humidity rules cover all possible values (completeness) to deduce high level abstractions, and the overlapping (correctness) between different works is resolved. Regarding the sun position elevation sensor, correctness and completeness are not satisfied yet, but we have 8 M3 rules to interpret the measurements.

In the M3 nomenclature[20], for each sensor, we implement M3 rules if we found them in the works referenced in the LOV4IoT dataset. When we implement the M3 rules we have to check manually completeness and correctness. A new M3 rule should not overlap with the previous M3 rules. For each sensor, we would like to have M3 rules covering all possible values, to get high-level abstractions in all cases. Moreover, sometimes two different works propose non compatible rules related to the same sensor. In this case, we choose the work

[20]http://www.sensormeasurement.appspot.com/documentation/NomenclatureSensorData.pdf

having the more rules related to a specific sensor, and delete the previous M3 rules related to the same sensor. Indeed, if the work has more rule, we consider that the rules are more precise and we can differentiate more abstractions from sensor data.

4.8.2 Evaluating LOV4IoT

In section 1.5, we introduced 'Hypothesis 6: A dataset of ontology-based projects relevant for IoT can be exploited outside of the M3 framework'.

A LOV4IoT evaluation form[21] has been set up to be filled by users (see Figure 4.20 and 4.21). It has been filled by 9 persons, this process is still ongoing. This form demonstrates that this tremendous work of synthesization and classification of all of these projects is useful for other developers, researchers and not only designed for the M3 framework. It helps them for their state of the art or finding and reusing the existing ontologies. Sometimes the results are not always 100% when the question was not mandatory or when we added later a new question to get more information. The LOV4IoT evaluation form contains the following questions and results:

- Who are you? (see Figure 4.20.A). According to the results, the users are either: 33% Semantic Web of Things developers or 22% Internet of Things developers. It does not attains 100% since we added this question later.

- How did you find this tool? (see Figure 4.20.B). According to the results, 22% found the LOV4IOT tool thanks to search engines, 22% thanks to people who recommended this tool (it includes others), 11% thanks to the research articles, and 11% thanks to emails that we sent to ask people to share their domain knowledge.

- Domain ontologies that you are looking for? (see Figure 4.20.C). According to the results, 55% of users are interested in weather ontologies, 44% in health ontologies, 33% in smart home ontologies, 22% in security, emotion or food ontologies, and 11% in agriculture, tourism or transportation ontologies. This means that users are interested in all domains that we cover.

- Do you trust the results since we reference research articles? (see Figure 4.20.D). According to the results, 55% of users trust the LOV4IOT tool since we reference research articles, 22% are not convinced. We do not know how we can convince them better.

- In which information are you interested? (see Figure 4.21.E). According to the results, 55% of users are interested in research articles and ontology URL referenced, 33% in rules, and 22% in sensors or technologies used.

- Do you use this web page for your state-of-the art? (see Figure 4.21.F). According to the results, 55% of users answered yes frequently, 22% yes, and 22% no.

[21] `https://docs.google.com/forms/d/1BMuRMlvbUAFYORtDRtdNZMXnvVgMGmy3pvRMgByA2EU/viewform`

- In your further IoT application developments, do you think you will use again this web page? (see Figure 4.21.G). According to the results, 66% of users answered yes frequently, 11% yes, and 11% no.

- In general, do you think this web page is useful: (see Figure 4.21.H). According to the results, 66% of users answered yes frequently, 22% yes, and 0% no.

- Would you recommend this web page to other colleagues involved in ontology-based IoT development projects? (see Figure 4.21.I). According to the results, 77% of users answered yes frequently, 11% yes, and 0% no.

These results are really encouraging to update our dataset with new domain, add more ontologies, etc.

4.8.3 Evaluating M3 domain knowledge with semantic web methodologies

In section 1.5, we introduced 'Hypothesis 7: The knowledge base built to interpret IoT data encourages the interoperability of data and domains.'. We **evaluated the M3 interoperable domain knowledge** as recommended by Methontology [Fernández-López et al., 1997] and Ontology Development Guide [Noy et al., 2001]. Figure 4.4 shows that the evaluation has been done with semantic web tools such as Oops[22] [Poveda-Villalón et al., 2012b] [Poveda-Villalón et al., 2012a], TripleChecker[23], RDF Validator[24], Vapour [Berrueta et al., 2008] and SSN Validator [Kolozali et al., 2014]. Oops detects some of the most common pitfalls appearing when developing ontologies. TripleChecker checks that we use common namespaces and ontologies and the appropriate concepts and properties. This tool helps to find typos and common errors in RDF data as a RDF validator. Vapour checks URI dereferenceable and enables to test easily our ontologies on other semantic web tools. We suggested more than 27 domain ontologies to LOV. Thanks to them we discovered numerous bad practices. As a consequence we redesigned our own ontologies and datasets to be compliant with best practices. Moreover, ontologies and datasets have been used to build cross-domain IoT applications. The M3 ontology has been checked with Fact++ and Hermit under Protege, we did not have any errors. Further M3 rules have been designed according to the M3 ontologies and have been run with the Jena inference engine. The M3 ontology, domain ontologies and datasets have been defined according to the best practices designed by knowledge engineering and ontology modeling. We employed all mentioned tools for our different domain knowledge that we redesigned such as naturopathy.

4.8.4 Discussions

The evaluation of the rule dataset validates 'Hypothesis 5: The dataset of M3 rules is reliable to interpret IoT data'. However, since we frequently add new sensors, actuators

[22] http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp
[23] http://graphite.ecs.soton.ac.uk/checker/
[24] http://www.w3.org/RDF/Validator/

Figure 4.20: Evaluating LOV4IoT through a user form (1)

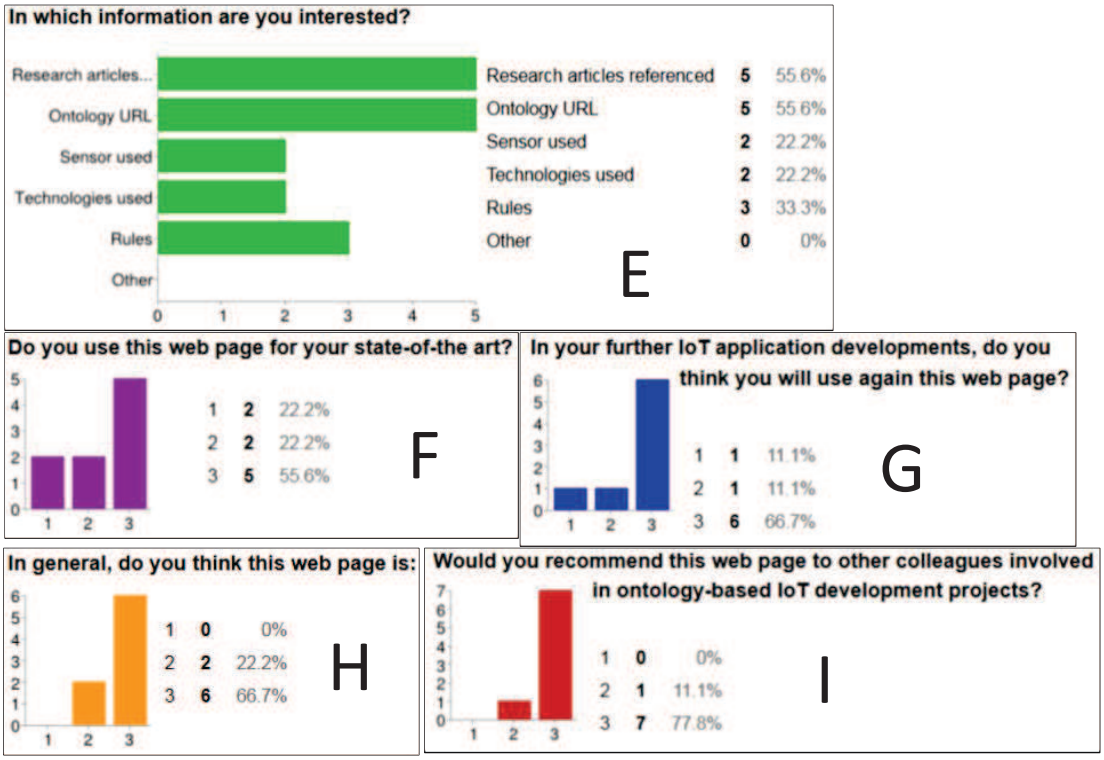

Figure 4.21: Evaluating LOV4IoT through a user form (2)

and domains to the M3 nomenclature, we have to find at the same time the related rules and check the completeness and correctness. This task is not always easy.

The evaluation of LOV4IoT with the user form shows that it validates 'Hypothesis 6: A dataset of ontology-based projects relevant for IoT can be exploited outside of the M3 framework'. This is really encouraging to maintain and improve this dataset. We are enriching it with new domains such as smart agriculture, smart city and smart energy. We would like to add filter to this dataset, according to the user's needs. Further, as future work, we would like to design methodologies to design the perfect ontology to be compliant with the M3 interoperable domain knowledge. We also want to exploit this dataset for the automatic extraction of the domain knowledge.

Regarding the evaluation of the M3 domain knowledge, it validates 'Hypothesis 7: The knowledge base built to interpret IoT data encourages the interoperability of data and domains'. However, we came to the conclusion that following semantic web best practices are not sufficient for reusing and combining domain knowledge due to heterogeneous issues. Indeed, the domain knowledge has been created by different stakeholders for different purposes. We tried to follow as much as possible best practices, but it is not always possible to follow them due to technical incompatibilities and lack of time. For instance, we wanted to have dereferenceable URIs, but because of our Google-based implementation, the automatic

Domain knowledge (ontology + dataset)	M3	Naturopathy	Health	Tourism	Transport	Weather	Home
RDF Validator	OK Tested 17/09/14						
Vapour	OK Tested 17/09/14						
TripleChecker	Error loading We discussed with Christopher Gutteridge (26/03/2014) Get the content to be "application/rdf+xml" Strange because it is working with other ontologies	Last tested 17/09/14: 2 issues avec foaf homepage document namespace not loaded	Draft (used for scenarios, we would like to build them with the automatic extraction)				
Linked Open Vocabularies (LOV) For ontologies	Issue with equivalent class with non LOV-able ontologies (we should delete them and replace them by seeAlso)	Suggested on LOV 06/05/2014 Issue we are not physicians in this domain TO DO: add all references that we read to build it	Draft (used for scenarios, we would like to build them with the automatic extraction)				
Linked Open Data (LOD) (For datasets)	Not done (small and simulated datasets)	Updated on LOD 05/09/2013	Draft (used for scenarios, we would like to build them with the automatic extraction)				
Oops	We fix some issues and avoid it for the other ontologies						
SSN Validation	Tested 19/09/14	Not relevant (these domain ontologies are not based on SSN)					

Table 4.4: Evaluate the M3 domain knowledge with semantic web tools

redirection was forbidden[25]. We also wanted to use PURL to change the location of our ontologies or datasets as we want. These two examples show the difficulties to be compliant with the best practices due to technical incompatibilities issues or lack of time.

4.9 Concluding Remarks

Today, most of the domain ontologies relevant for IoT: (1) are not published online and cannot be reused, (2) do not follow semantic web best practices, and (3) are not interlinked

[25] `https://cloud.google.com/appengine/docs/java/config/webxml`

with each other. We have built the LOV4IoT dataset to synthesize and classify more than 270 ontology-based projects, which could be exploited to enrich sensor measurements. These ontologies could be better reused if best practices regarding vocabulary publishing would have been enforced. The LOV4IoT dataset was highly employed to build the M3 ontology and the M3 interoperable domain knowledge. The LOV4IoT dataset was a necessary step to build S-LOR, a logic-based approach to deduce high-level abstractions from sensor data. The main novelty of 'Sensor-based Linked Open Rules' approach that it is possible to share and reuse rules applied to sensors, as it has been done previously for the Linked Open Data and the Linked Open Vocabularies. The second main advantage of S-LOR is that, we can easily combine heterogeneous domains to provide cross-domain suggestions.

During this thesis, we have shared the lessons learned by disseminating our work in conferences but also standardizations and remind to the next ontology designers the semantic web best practices that we acquired. Actually, there is a real need to popularize semantic web best practices and standardize common descriptions as we explained in W3C Web of Things [Gyrard et al., 2014a], ETSI M2M [Gyrard et al., 2013] and OneM2M [Gyrard and Bonnet, 2014] standardizations. All semantic web bad practices encountered and the related guidelines to remedy them are available in a draft document written [Gyrard and Bonnet, 2014] for the OneM2M international standard. Reusing the existing domain knowledge and linking them is not so easy.

Future work is enriching S-LOR with more complex sensors and rules to interpret electrocardiogram (ECG), accelerometer or gyroscope data measurements or activity rules. An interesting task would be to integrate to S-LOR the KAT toolkit devised by [Ganz et al., 2013] [Ganz et al., 2014] and adapt it if necessary. Another interesting task would be to integrate a semantic and rule-based recommender system to adapt the M3 suggestions to the user profile. For instance, we could reuse and extend cross-domain recommender systems designed by [Hoxha, 2014] or [Tobías, 2013]. Regarding ontology matching or ontology and rule editors, we could improve existing tools, combine them if needed and adapt them to the LOV4IoT dataset.

In the next chapter (Chapter 5), we explain how to aid developers to secure their applications.

Chapter 5

Security Toolbox: Attacks & Countermeasures (STAC)

"When our body-mind is in concert with the universe, everything becomes spontaneous and effortless."

Deepak Chopra

In this chapter, we assist IoT developers to design secure IoT applications by addressing "Challenge E: Securing IoT applications". We assume in this work that the developers want to secure their applications. Securing Internet of Things and Machine to Machine (M2M) applications or architectures is a challenging task because there is a need to secure heterogeneous communications, devices and applications. It is really time-consuming and not an easy task to learn security in all of these topics and choose the right security mechanisms fitting the needs. To achieve this challenge, we have designed the STAC (Security Toolbox: Attacks & Countermeasures) to assist software developers and designers who are not expert in security to find the right security mechanisms to secure Internet of Things (IoT) applications.

This chapter is composed as follows. Section 5.1 introduces STAC components to assist developers in securing IoT applications. The STAC approach is inspired from the M3 approach. Section 5.2 explains the STAC generator to guide users to secure applications. Section 5.3 describes the STAC ontology and dataset which have been designed using the approach employed to build the M3 interoperable domain knowledge (explained in section 4.4). Section 5.4 provides a proof-of concept. Section 5.5 evaluates STAC. Section 5.6 reminds the novelty of STAC. Section 5.7 concludes this chapter and outlines future work.

In this chapter, we assess the following research questions:

- How to assist developers, using a 'security by design' approach, in securing IoT applications?
- How to secure IoT architectures and applications since they employ various technologies?

- How to reuse existing security ontologies?
- How to provide a cross-domain security knowledge base?
- How to conceive a knowledge base following the semantic web best practices?
- How to reuse the M3 approach in the security domain?

5.1 Assisting Developers in Securing IoT Applications

In Figure 5.1, the developers through the STAC template generator use the STAC security knowledge base to find attacks and security mechanisms specific to the technologies used in an application. For each security mechanism or attack, they obtain additional information such as security properties (e.g., authentication and confidentiality) satisfied, the OSI model layer involved and the advantages and drawbacks of a security mechanism. STAC has been build using the M3 approach as it is shown in Figure 5.1:

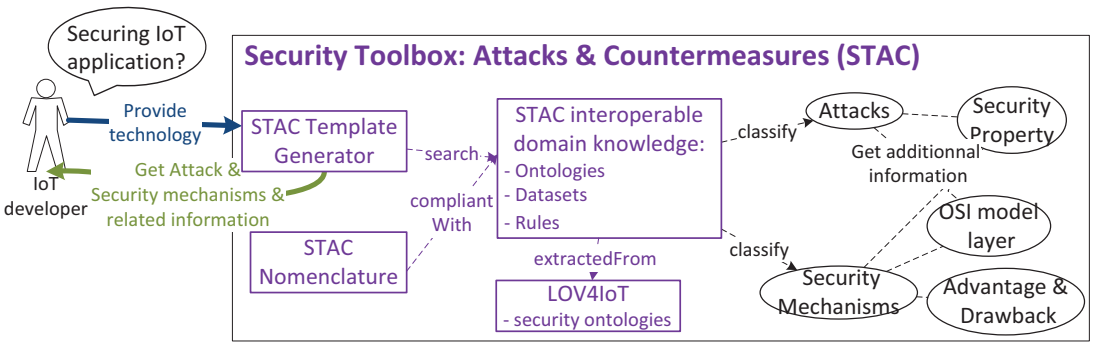

Figure 5.1: Assisting developers in securing IoT Applications with STAC

- The STAC generator enables finding attacks and security mechanisms according to a specific technology. We also designed some templates based on existing projects involving sensors. For instance, the CodeBlue project [Malan et al., 2004] employs sensors such as pulse oxymeter, Electrocardiogram (ECG) and the Wi-Fi technology. They explain in their work that they use the Elliptic Curve Cryptography (ECC) cryptography algorithm to encrypt sensor data. This information has been integrated in a STAC template to later help developers to secure similar applications.
- The STAC nomenclature to use common terms and avoid any ambiguities. For instance, a beginner in security or a machine does not know explicitly that a asymmetric algorithm is a synonym for public key algorithm. For this reason, we define common terms to ease interoperability between existing works. This step is essential to design the security cross-domain knowledge base.

- Security ontologies have been studied and classified in the LOV4IoT dataset and reused to build the STAC interoperable knowledge base.
- The STAC security knowledge base has been built using the same approach for building M3 interoperable domain knowledge. Further, it enables to combine several security domains such as sensor networks, wireless networks cellular networks, network management, web applications, etc.

5.2 STAC generator

Figure 5.2 shows a STAC template defining security mechanisms already used in sensor-based projects. For instance, in the TALISMAN + EU project [Hervás et al., 2013], they used the secure Socket Layer (SSL) security mechanism and the X_509 certificates to secure their applications, more precisely to secure the transmission of their sensor data generated by blood pressure and cholesterol sensors.

This STAC template is described in the LOV4IoT RDF dataset[1].

```
<m3:M2MApplication rdf:about="Bravo">
    <m3:hasContext rdf:resource="&m3;Health"/>
    <rdfs:label xml:lang="en">[Bravo et al. 2009-2013]. See LOV4IoT for more details.</rdfs:label>
    <rdfs:comment xml:lang="en">Paper: Mobile monitoring and reasoning methods to prevent cardiovascular
    <m3:hasM2MDevice rdf:resource="&m3;BloodPressureSensor"/>                diseases 2013</rdfs:comment>
    <m3:hasM2MDevice rdf:resource="&m3;CholesterolSensor"/>
    <m3:hasSecurityMechanism rdf:resource="&stac_data;SSL"/>
    <m3:hasSecurityMechanism rdf:resource="&stac_data;X_509"/>
    <dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2009</dcterms:issued>
    <lov4iot:hasOntologyStatus rdf:resource="&lov4iot;WaitForAnswer"/>
</m3:M2MApplication>
```

Security mechanisms suggested

Figure 5.2: STAC RDF template securing health applications with SSL and X_509

The STAC template is also available through a user interface as depicted in Figure 5.3. For instance, the users want to secure a health application using a cholesterol sensor. Users search in the drop-down list the sensor that they looking for. The SPARQL query will ask the LOV4IoT RDF dataset and returns the result displayed in Figure 5.3: to secure data provided by this sensor, users can use the SSL and X_509 security mechanisms. The user interface is the same that we used for S-LOR[2].

5.3 Interoperable STAC Cross-Domain Knowledge

Security concerns should always be addressed when designing applications. We have built the STAC security knowledge base to assist software developers and designers in securing IoT/M2M architectures and IoT/M2M applications, etc. However, STAC can be used to secure any kind of applications.

[1] www.sensormeasurement.appspot.com/dataset/lov4iot-dataset

[2] http://www.sensormeasurement.appspot.com/?p=swot_template

Figure 5.3: STAC user-interface template securing health applications with SSL and X_509

5.3.1 Reusing Security Knowledge with LOV4IoT

We found, classified and exploited more than 24 ontology-based works in the LOV4IoT dataset related to sensor networks, mobile phones, cellular networks, Intrusion Detection System (IDS) and cryptography as it has been presented in section 2.4.4. Due to their numerous limitations, we redesigned an interoperable security knowledge base called STAC (Security Toolbox: Attacks & Countermeasures) [Gyrard et al., 2013] as it has been done for the M3 interoperable domain knowledge to interpret sensor data. We encountered the same issues concerning the security expertise such as lack of semantic web best practices, ontologies not shared online, heterogeneous terms to describe the same concept and ontologies structured in different ways, etc. The STAC knowledge base enables to link several security domains together as depicted in Figure 5.4.

5.3.2 STAC ontology

The STAC ontology[3] describes the main security and cryptographic concepts and mechanisms used in various domains: sensor networks, cellular networks (2G, 3G, 4G), wireless networks (Wi-Fi, Bluetooth, Wimax, Zigbee, Mesh, M2M, Manet, RFID), web applications, Intrusion Detection Systems (IDS) and network management. The main goal of this ontology is to suggest the best security mechanisms to design secure applications. To achieve this goal, we have designed the STAC ontology specifying the relationships between several concepts as depicted in Figure 5.5:

- `Attack` which explains vulnerabilities in a technology. For instance, the jamming attack is common to all wireless technologies.
- `SecurityMechanism` which describes solutions to secure applications.
- `Technology` which classifies sensor networks, cellular networks (2G, 3G, 4G), wireless networks (Wi-Fi, Bluetooth, Wimax, Zigbee, Mesh, M2M, Manet, RFID), web applications, Intrusion Detection Systems (IDS) and network management.

[3] `http://securitytoolbox.appspot.com/stac#`

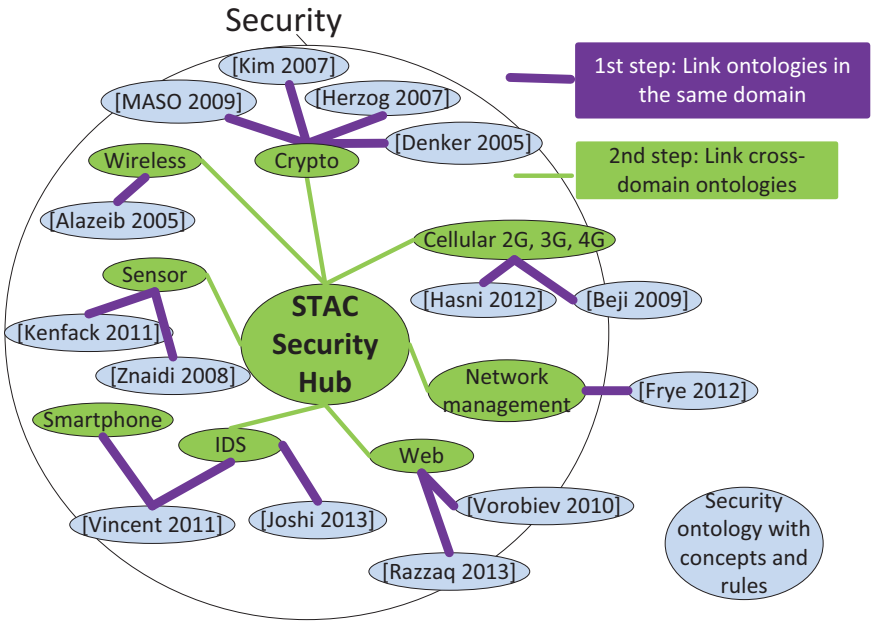

Figure 5.4: The STAC knowledge base

- `SecurityProperty`. For instance, authentication,integrity and confidentiality are well-known security properties.

- `OSIModel` has been defined to classify attacks and security mechanism in each layer of the communication system stack.

- `Feature` which is used to indicate advantages and drawbacks of the security mechanisms. For instance, 'deprecated' is considered as a drawback. Security mechanisms having this feature should not be used anymore to secure systems.

In the STAC ontology, we link common security concepts (e.g., `EncryptionAlgorithm`) to other existing security ontologies published online and presented in section 2.4.4.

Technology

A `Technology` is vulnerable to `Attack` (`hasVulnerability` property) and has a specific `SecurityMechanism` (`isProtectedBy` property). For example, all wireless technologies have the `Jamming` attack in common due to the wireless communication, which is not the case for wired networks. In STAC, we focused on several technologies and the related instances such as `NetworkManagement, Web (ProgrammingLanguage, Ecommerce, Frameworks, Databases)`, wired (`Ethernet`) and wireless networks: `SensorNetwork, M2M, Wi-Fi, GSM (2G), UMTS (3G), LTE (4G)`, etc. (see Figure 5.6).

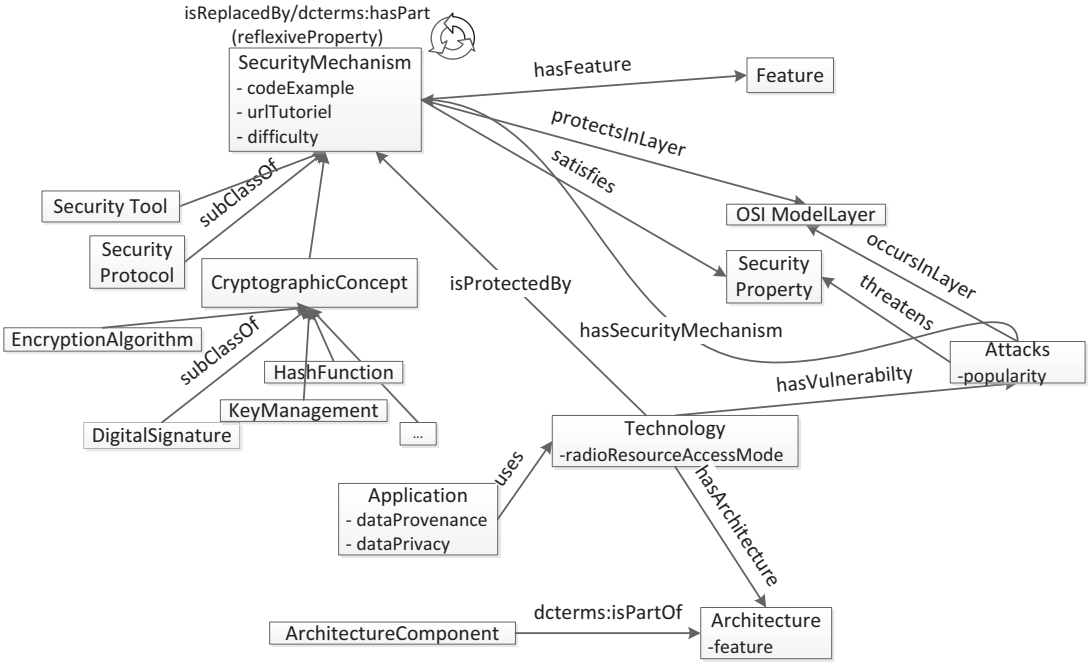

Figure 5.5: The top level part of the STAC ontology

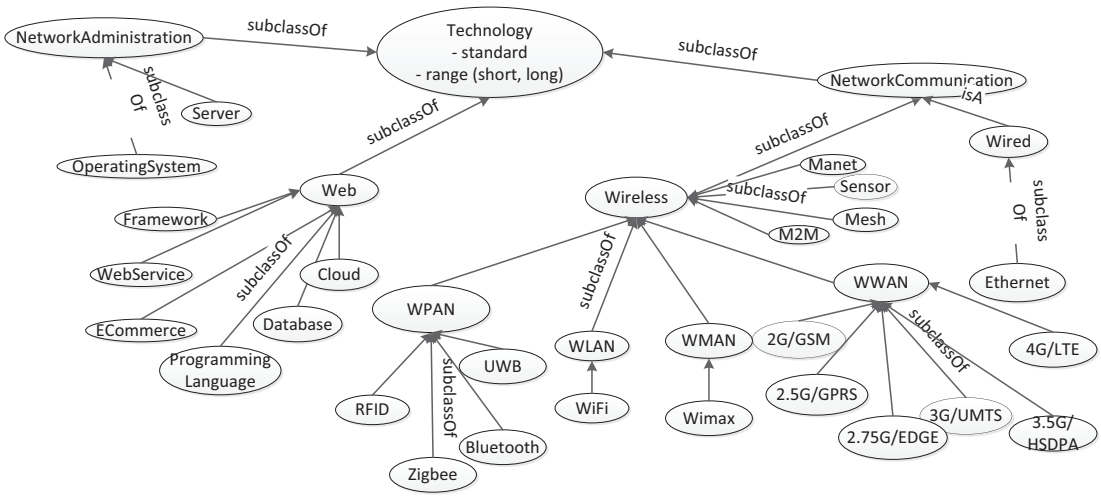

Figure 5.6: The technology concept and its subclasses

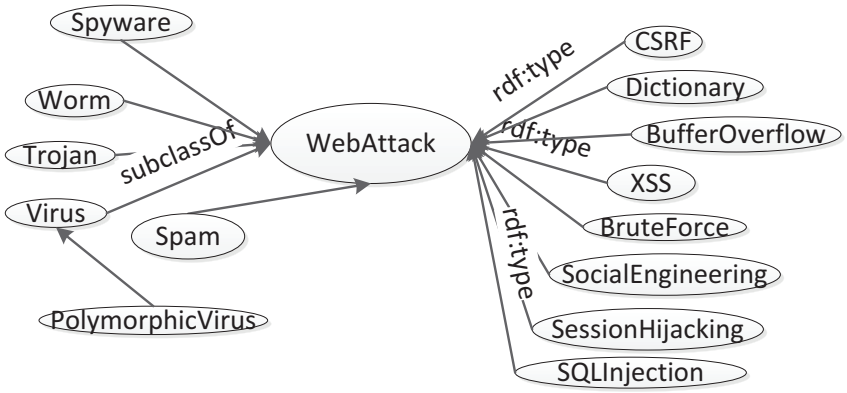

Figure 5.7: The web attack subclasses and instances

A `Technology` can be replaced by another technology that is more recent (`isReplacedBy` property). This is the case for cellular technologies where the GSM technology has been replaced by the GPRS technology.

Attack

We classify `Attack` according to the `OSIModelLayer` and the `Technology`. For example, the `Jamming` attack occurs in the `PhysicalLayer` and is specific to `SensorNetwork` whereas the `SQLInjection` occurs in the `ApplicationLayer` and targets `Web` applications. We have referenced numerous technologies, attacks and security mechanisms according to the OSI model.

In the STAC ontology, we specify restrictions between attacks and the security mechanisms for each technology. For example, `SensorAttack` can exclusively be protected by `SensorSecurityMechanism` and `WebAttack` by `WebSecurityMechanism`: the VPN (Virtual Private Network) security mechanism is a web security mechanism which cannot thwart sensor attacks.

In Figure 5.7, we have classified the attacks that are specific to web technologies. Some web attacks are defined as concepts in the STAC ontology such as Worm to add specific worms such as Morris in the dataset.

OSI model

The `OSIModel` concept is a collection of seven `OSIModelLayer` concepts:

- `Physical Layer` which defines the electrical and physical specifications of the data connection.
- `Link Layer` which is responsible for a reliable link between two nodes, by detecting and correcting errors that may occur in the physical layer.

- `Network Layer` which handles the addressing and routing of data.
- `Transport Layer` which manages packetization of data, then the delivery of the packets, including checking for errors in the data once it arrives.
- `Session Layer` which controls the dialogues (connections) between computers.
- `Presentation Layer` which is usually part of an operating system (OS) and converts incoming and outgoing data from one presentation format to another.
- `Application Layer` which is the layer closest to end users. It means that both the OSI application layer and the user interacts directly with software applications.

The OSI model concept enables classifying instances of attacks and security mechanisms according to the OSI model layers. For instance, the buffer overflow attack is specific to the application layer.

Security property

The security property concept (`SecurityProperty`) indicates the security properties: (1) ensured by each security mechanism (`satisfies`), and (2) threatened by each attack (`threatens`). Let's take an example to see how these three concepts are linked. The Virtual Private Network (VPN) security mechanism satisfies the authentication, confidentiality and integrity properties and that the social engineering attack threatens the authentication property.

In our ontology, we have identified twelve security properties:

- `Confidentiality` keeps information secret from eavesdroppers and unauthorized parties.
- `Authentication` ensures that a receiver is capable of identifying the authenticity of the message.
- `Integrity` ensures that a message is not altered in transit.
- `Access Control` defines who has access to which document.
- `Non Repudiation` provides proof of: (1) the integrity and origin of data, and (2) an authentication that can be asserted to be genuine with high assurance.
- `Date Freshness` or replay protection checks that the data is recent and an adversary has not replayed old messages.
- `Availability` ensures that the network is alive and that data is accessible. It means that if an attack succeeds, its impact should be minimized. The compromise of a single node should not break the security of the entire network. It also ensures that services are available in any case.
- `Semantic Security` ensures that an eavesdropper has no information about the plaintext, even he it sees multiple encryptions of the same plaintext.

- `Trust` checks if the data is reliable.
- `Anonymity` protects user identity, making it hard to track the whereabouts of a certain user.
- `Authorization` ensures that only authorized persons can access to network services or resources.
- `Accountability` is the requirement that actions of an entity may be traced uniquely to that entity.

Security mechanism

The security mechanism concept (`SecurityMechanism`) has been defined to identify the security mechanisms that can be used to protect `Application` against specific `Attack` as depicted in Figure 5.8. It can be:

- Security tools (`SecurityTool`) such as a network security tool, message encryption tool, proxy, sniffer, etc.
- Security protocols (`SecurityProtocol`) such as web security protocol, sensor security protocol, WiFi security protocol.
- Cryptographic concepts (`CryptographyConcept`) such as hash function, digital signature, key management, asymmetric algorithms and symmetric algorithms (block cipher or stream cipher).

A `SecurityMechanism` can be itself composed of other security mechanisms. For example, the VPN security mechanism is composed of (`dcterms:hasPart` property) the Internet Key Exchange (IKE) key management and the IPSec protocol which are both security mechanisms. Technologies are protected by specific security mechanisms. Indeed, sensor security mechanisms are devoted to secure the sensor technology, Wi-Fi security mechanisms protect Wi-Fi technologies, etc.

As security mechanism can be replaced by another more secure mechanism, we have added the property `isReplacedBy`. For example, for the Wi-Fi technology, the WEP security mechanism has been replaced by WPA1 which has been replaced by WPA2.

Feature

To assist developers in choosing the best security mechanism among all the existing mechanisms, there is a need to differentiate them by indicating their strengths and weaknesses. We have created the concept called `Feature` to fulfill this need. The `Feature` concept is composed of several properties such as `Free`, `Flexible`, `Scalable`, `Secured`, `LowCostDeployment`, `LowEnergyConsuming`, `ExchangeKeyEasy` and `SuitableHeterogeneousCommunication`. Hence, we can indicate that an asymmetric algorithm is high energy consuming, but propose an easy solution to exchange keys. A symmetric algorithm is low energy consuming,

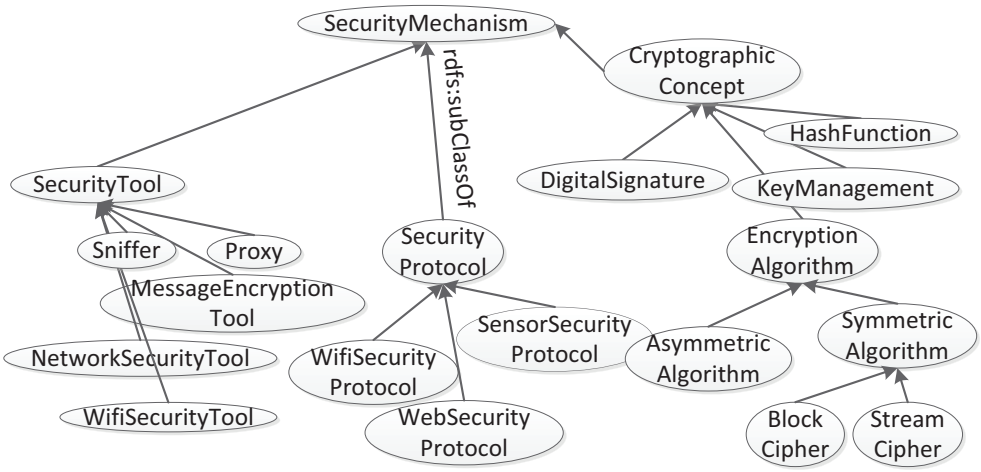

Figure 5.8: Security mechanism subclasses

however, exchanging the keys is not an easy task. Another example is the difficult task to secure communications due to various protocols: there are three main security protocols to secure Wi-Fi communications: WEP, WPA1 and WPA2. The latter is the most secure security mechanism.

5.3.3 STAC dataset

The STAC ontology has been used to build the STAC dataset[4] to reference numerous attacks and security mechanisms in heterogeneous domains. We focused on the security for sensor networks, but it would be nice to add other sections in the same way for the other domains that we implemented in the STAC dataset such as 2G, 3G, 4G, WiFi, RFID, Zigbee Bluetooth, Wimax, Mesh, M2M. The state of the art of such technologies have been partially covered.

Security for sensor networks

To build the dataset for security of sensor networks, we have studied the existing attacks and security mechanisms defined specifically for sensor networks [Boyle and Newe, 2007] [Ahmed, 2009] [Borgohain et al., 2015] [Karlof et al., 2004] [Perrig et al., 2002] [Lighfoot et al., 2007] [Luk et al., 2007] [Zhu et al., 2006] [Casado and Tsigas, 2009] [Douceur, 2002] [Hamid et al., 2006]. In this dataset, we provide all the security attacks and security mechanisms (security protocols and key management) specific to sensor networks, that we have classified according to the OSI model (Figure 5.1). The security mechanisms specific to the link layer are

[4] `http://securitytoolbox.appspot.com/stac-dataset`

OSI Model Layer	Attacks	Security Mechanisms
Application Layer	No	No
Presentation Layer	No	No
Session Layer	No	No
Transport Layer	DoS, Flooding, Desynchronisation	No
Network Layer	Sybil, Wormhole, Sinkhole, Hello Floods, spoofing, selective forwarding, etc.	MiniSec, LEAP, ContikiSec
Link Layer	Collision, Unfairness, exhaustion	TinySec, LLSP, SPINS (SNEP, TESLA)
Physical Layer	Jamming, Tampering	No
Unclassified	No	LiSP

Table 5.1: Classification of attacks and security mechanisms specific to sensor networks according to the OSI model

TinySec [Karlof et al., 2004], SPINS [Perrig et al., 2002] and LLSP [Lighfoot et al., 2007] which can thwart sensor attacks such as collision, unfairness and exhaustion. Regarding the network layer, we considered the MiniSec [Luk et al., 2007], LEAP [Zhu et al., 2006] and ContikiSec [Casado and Tsigas, 2009] security mechanisms which can thwart sensor attacks such as Sybil [Douceur, 2002], Wormhole, Sinkhole, Hello Floods [Hamid et al., 2006], etc. All these security protocols use specific cryptographic algorithms and satisfy security properties. However, these algorithms or security properties are not always specified for each security protocol. When the algorithms or security properties are provided, we have added them in the dataset as it is shown in Figure 5.2. As an example, the description of the LLSP security mechanism has been implemented in the STAC dataset as an instance (see Figure 5.9), the cryptographic algorithms used are CBC and AES, the security properties satisfied are data freshness, confidentiality, authentication and semantic security.

Security for 3G

As it has been done for sensor networks, we build the dataset for security of 3G, more precisely of Universal Mobile Telecommunications System (UMTS). We have studied the existing attacks and security mechanisms defined specifically for 3G (see 5.3): [Al-Massari, 2009] [Mobarhan et al., 2012] [Meyer and Wetzel, 2004] [Xenakis and Merakos, 2004] [Al-Saraireh et al., 2006] [Koien, 2004] [Abid et al., 2002] [Kitsos et al., 2007] [Boman et al., 2002] [Caragata et al., 2011b] [Caragata et al., 2011a] [Ahmadian et al., 2009].

Security for 4G

As it has been done for sensor networks, we build the dataset for security of 4G. We have studied the existing attacks and security mechanisms defined specifically for 4G: [Park and Park, 2007] [Gupta and Patil, 2009] [Seddigh et al., 2010] [Aiash et al., 2010] [Rana, 2011] [Rahman and Sharma, 2012] [Abdelkader, 2009].

Security mechanism / Security Property	TinySec	SPINS		LLSP	MiniSec	LEAP	ContikiSec
		µTESLA	SNEP				
Confidentiality	Yes (Skipjack)	No	Yes (Encryption)	Yes (AES, CBC, CTR)	Yes (Skipjack)	Yes	Yes (AES)
Authenticity	Yes (CBC-MAC)	No	Yes (MAC)	Yes (MAC)	Yes (OCB)	Yes	Yes
Integrity	Yes (MAC)	No	Yes (MAC)	Yes (MAC)	No	No	Yes
Freshness/ Replay protection	Yes	No	Yes (Weak / MAC)	Yes	Yes	No	No
Access Control	No	No	No	Yes	No	No	No
Non Repudiation	No	No	No	No	No	No	No
Semantic Security	Yes (Initial Vector)	No	Yes	Yes	No	No	No

Table 5.2: Security properties satisfied for sensor security mechanisms

```
<stac:SensorSecurityProtocol rdf:about="LLSP">   => Security mechanism
    <rdfs:label xml:lang="en">Link-Layer Security Protocol (LLSP)</rdfs:label>
    <rdfs:comment xml:lang="en"></rdfs:comment>
    <dc:description xml:lang="en">See Paper: An energy efficient Link-Layer Security
    Protocol for wireless sensor networks [Lighfoot et al. 2004]</dc:description>
    <stac:protectsInLayer rdf:resource="LinkLayer"/>
    <dcterms:hasPart rdf:resource="CBC"/>
    <dcterms:hasPart rdf:resource="AES"/>
    <stac:satisfies rdf:resource="DataFreshness"/>
    <stac:satisfies rdf:resource="Confidentiality"/>
    <stac:satisfies rdf:resource="Authentication"/>
    <stac:satisfies rdf:resource="SemanticSecurity"/>
</stac:SensorSecurityProtocol>
```

=> Cryptographic algorithms used

=> Security properties satisfied

Figure 5.9: The description of the LLSP security mechanism described in RDF

3G Security protocol	AKA, EAP-AKA
3G Cryptographic algorithms	MILENAGE, f1, f2, f3, f4, f5, f8, f9
3G Attacks	Hijack IMSI, MITM, Sniffing, Social Engineering, Spoofing IP/MAC address, Collision, DoS, Eavesdropping

Table 5.3: Security attacks and security mechanisms for 3G

```
<owl:Class rdf:ID="BluetoothTechnology">
    <rdfs:label xml:lang="en">Bluetooth Technology</rdfs:label>
    <rdfs:comment xml:lang="en">A protocol for short-range (up to 100 meters)
    <rdfs:subClassOf rdf:resource="#Technology"/> wireless networks.</rdfs:comment>
    <rdfs:seeAlso rdf:resource="&serviceContext;Bluetooth"/>
    <rdfs:seeAlso rdf:resource="&ct;BluetoothConnectivity"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasVulnerability"/>
            <owl:someValuesFrom rdf:resource="#BluetoothAttack"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 5.10: Adding a new technology in the STAC ontology

The same work has been partially achieved for the these technologies: cellular networks (2G), wireless networks (Wi-Fi, Bluetooth, Wimax, Zigbee, Mesh, M2M, Manet, RFID), web applications, Intrusion Detection Systems (IDS) and network management.

5.3.4 Updating STAC

Updating the STAC ontology or dataset is really easy. For instance, in Figure 5.10, we added a new technology, the Bluetooth technology. When we add a new technology in the ontology, it is recommended to add the related attacks and the security mechanisms concepts specific to this technology and the related restrictions.

5.4 Implementation

The implementation has been done with the same technologies employed for the M3 framework. The STAC application[5] proposes a menu composed of:

- The cryptography user interface proposing the encryption algorithms, hash functions, digital signatures, mode of operations and key management (see Figure 5.11).
- The security property user interface containing a list of the security properties and their methods.
- The attacks and security mechanisms user interface containing the threats, and for each threat the security mechanisms suggested to protect against it. In the user interface, the attacks and security mechanisms are listed according to the OSI model and the technologies (Figure 5.13).

[5] `http://www.sensormeasurement.appspot.com/index.html?p=stac`

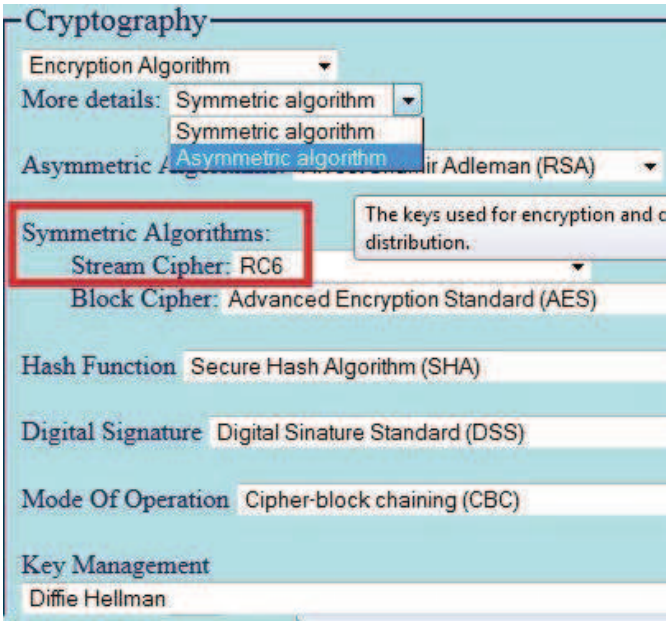

Figure 5.11: The cryptography user interface

- The security interface specific for each communication networks containing the security protocols, cryptographic algorithms and attacks. Communication networks can be: (1) cellular technologies such as 2G, 3G, 4G, or (2) wireless technologies such as sensor networks, Wi-Fi, Bluetooth, Wimax, Machine-to-Machine (M2M) and Mesh networks. Figure 5.12 show the security for 2G.

- The STAC user interface suggesting all attacks and security mechanisms specific to a technology. An example is given in the use case presented in section 6.6.

All drop-down lists displayed in these user interfaces interact with the STAC security knowledge base via SPARQL queries, and then display the obtained results.

Figure 5.11 shows the main cryptographic concepts. It explains that an encryption algorithm is either a symmetric or asymmetric, and the tooltip teaches that the keys used in an asymmetric algorithm are different for encryption and decryption. An instance of an asymmetric algorithm can be RSA. Symmetric algorithms can be either stream cipher (e.g., RC6) or block cipher (e.g., AES). The interface displays also hash functions (SHA), digital signatures (DSS), mode of operation (CBC) and key management (Diffie Hellman) by using a drop-down list.

The interface depicted in Figure 5.13 allows displaying all attacks and proposes the right security mechanisms to thwart them. For example, to thwart the eavesdropping attack, we propose the HTTPS security mechanism. A click on the drop-down list also proposes authentication method, directional antenna, encryption algorithm, and the VPN security mechanisms. We also indicate for each security mechanism the security properties satisfied

Figure 5.12: The 2G cellular network user interface

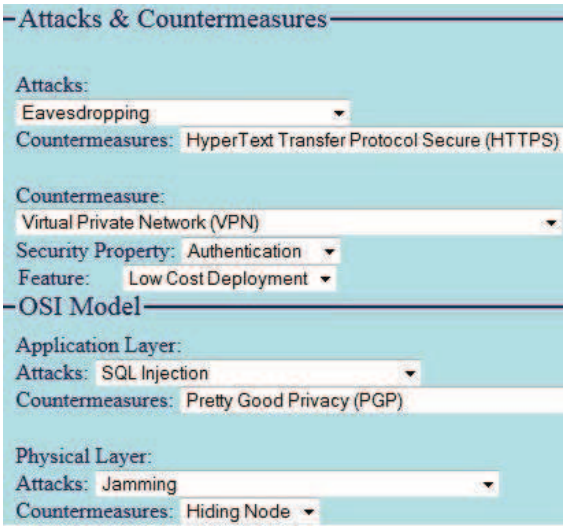

Figure 5.13: The attacks and security mechanisms user interface

and their features. The VPN satisfies the authentication, integrity, confidentiality, access control, privacy and authorization properties and its features are `LowCostDeployment` and `Secured`. In the OSI model section are classified all attacks and all security mechanisms according to the OSI model layer. For example, the SQL injection occurs in the application layer, the PGP security mechanism protects the application layer, etc.

Domain knowledge (ontology + dataset)	RDF Validator	Vapour	TripleChecker	Linked Open Vocabularies (LOV)	Linked Open Data (LOD)	Oops
STAC (Security)	Ok Last tested 17/09/14	2/3 tests ok Last tested 17/09/14 Issue: Content type should be 'application/rdf+xml'	Ok Last tested 17/09/14	1st ontology related to the security domain on LOV 03/05/2013	Updated on LOD 14/05/2013	Yes

Figure 5.14: STAC has been evaluated with the semantic web tools

5.5 Evaluation

Firstly, we have evaluated STAC with semantic web methodologies and tools as it has been done for interpreting IoT data to validate 'Hypothesis 8: The security knowledge base is built using the same methodology that for the M3 interoperable domain knowledge'. Then, we have evaluated it with end users to validate 'Hypothesis 9: A security knowledge base can help non-experts in security to choose security mechanisms fitting their needs to secure IoT applications'.

5.5.1 Evaluating STAC domain knowledge with semantic web methodologies

In section 1.5, we introduced 'Hypothesis 8: The security knowledge base is built using the same methodology that for the M3 interoperable domain knowledge.'. To evaluate our proposed approach, we have used the methodology and evaluation, as it has been done for the M3 interoperable domain knowledge, for the STAC knowledge base. We used the same semantic web tools such as Oops, TripleChecker, RDF validator and Vapour to evaluate and if required improve the STAC knowledge base (see Figure 5.14). One issue remains (content type should be application/rdf+xml) with Vapour which is difficult for us to solve since it requires network management skills or access rights on the server to fix this issue. Moreover, the STAC ontology and dataset have been evaluated and accepted by the semantic web community since the STAC ontology is now referenced by the LOV project. The LOV project added the security domain thanks to our work (see Figure 5.15). STAC has been referenced by LOV with some efforts. Indeed, we had several links to existing 'non perfect' security ontologies such as `owl:equivalentClass`. The LOV catalogue recommends us to replace those links by `rdfs:seeAlso` whereas those links were recommended by semantic web methodologies [Noy et al., 2001]. Indeed, security ontologies that we found on the Web hindered the automation of the LOV bots since URI are not dereferenceable, etc.

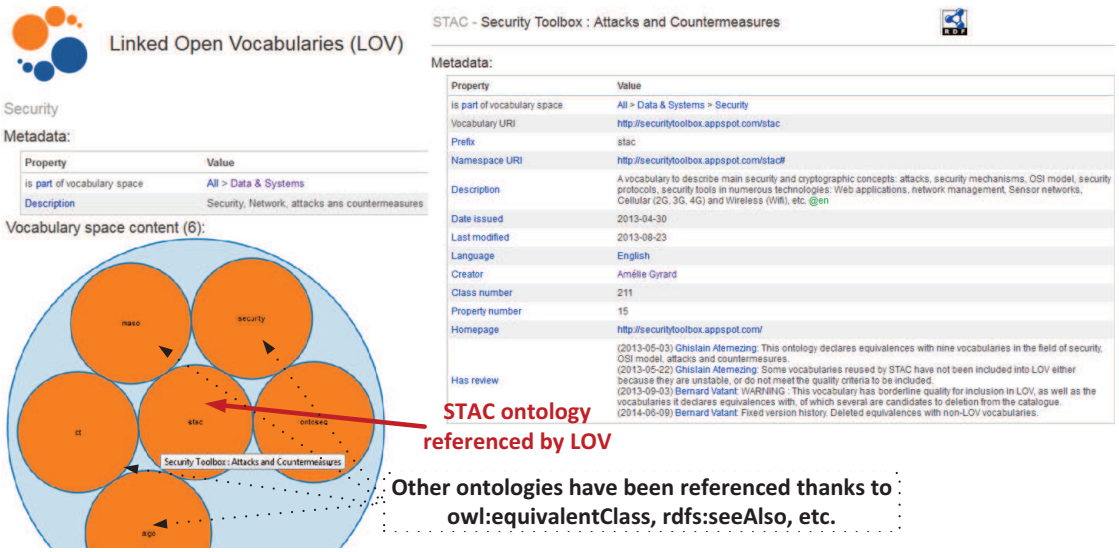

Figure 5.15: STAC referenced by the Linked Open Vocabularies catalogue

5.5.2 Evaluating STAC with end users

In section 1.5, we introduced 'Hypothesis 9: A security knowledge base can help non-experts in security to choose security mechanisms fitting their needs to secure IoT applications'. To evaluate our proposed approach, we have create an evaluation form[6] which has been filled by developers and researchers in computer sciences to evaluate whether the STAC knowledge base can assist them in securing their applications. The form contained the following questions (see Figure 5.16) and we obtained twenty eight responses[7] as follows:

- Your knowledge in security? (see Figure 5.16.A). According to the results, the STAC application has been tested by different kind of users: 4 % are not familiar with security at all, 46% of users has a low knowledge in security, 25% are intermediate in knowledge security and 21% are experts in security.
- Who are you? (see Figure 5.16.B). Finding IoT developers was difficult, so we sent this evaluation form to users who might be interested in securing their applications. According to the results, 36% are researchers and 36% are software developers. At the beginning filling this question was not mandatory, so the total is not 100%.
- Are the concepts intuitive and easy to understand? (see Figure 5.16.C). According to the results, 39% of users understand the concepts from the STAC knowledge base, this is higher than the number of security experts, so it means that non expert in security can understand security concepts thanks to the STAC knowledge base. However, 25%

[6] https://docs.google.com/forms/d/1NKiMQPVR6X6Reioud0-WBZu1bmo3T1Ah7PZm9De-apk/viewform

[7] https://docs.google.com/forms/d/1NKiMQPVR6X6Reioud0-WBZu1bmo3T1Ah7PZm9De-apk/viewanalytics

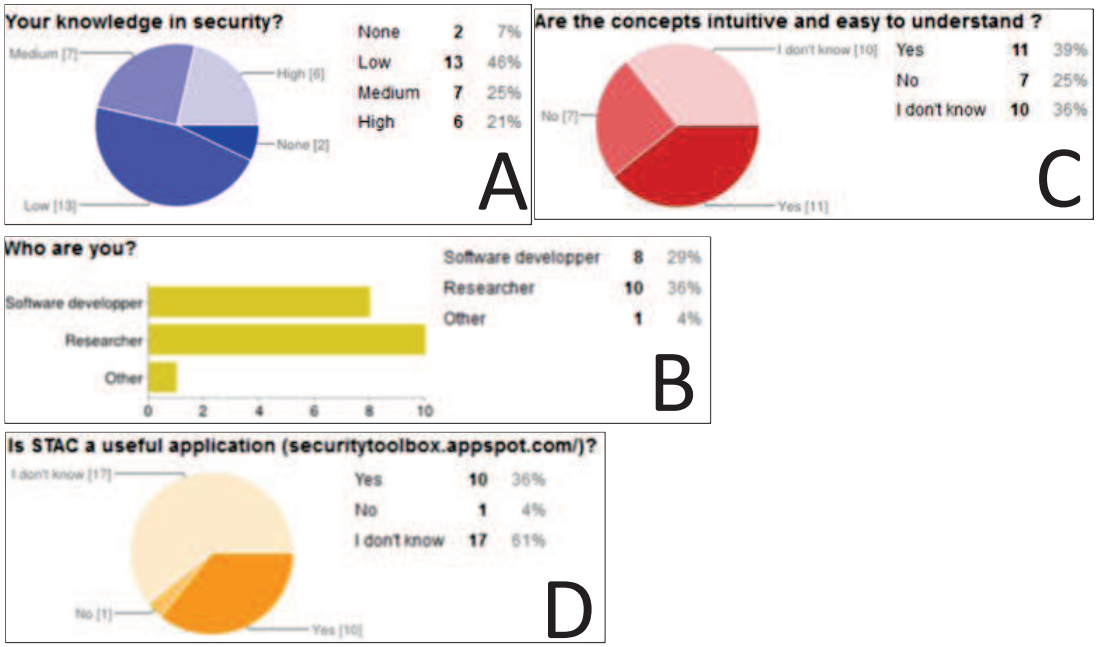

Figure 5.16: Evaluation form results from end users

do not understand the concepts at all and 36% do not know if they understand the security concepts. These results shows there is a need to vulgarize much more security concepts to non-security experts.

- Is STAC a useful application? (see Figure 5.16.D). According to the results, 36% found the STAC knowledge base useful, which is displayed in the user inferface, 4% not, and 61% did not well understand the usability. It means that the STAC application should be improved and we should try to hide security to users by automatically choosing the security mechanism fitting the need of the applications and even set up the security mechanisms.

In the same form, we asked the following questions to decide which new technologies should we integrate in STAC and related attacks and security mechanisms (see Figure 5.17):

- What kind of applications do you need to secure? (see Figure 5.17.A). Users need to secure web and mobile applications. STAC will be enriched with these new domains.

- Are you interested in security for wireless networks? (see Figure 5.17.B). Twenty users were interested to know security related to wireless networks, WiFi, 3G, 4G and sensor networks. We enriched STAC with these domains, the related security vulnerabilities and security mechanisms.

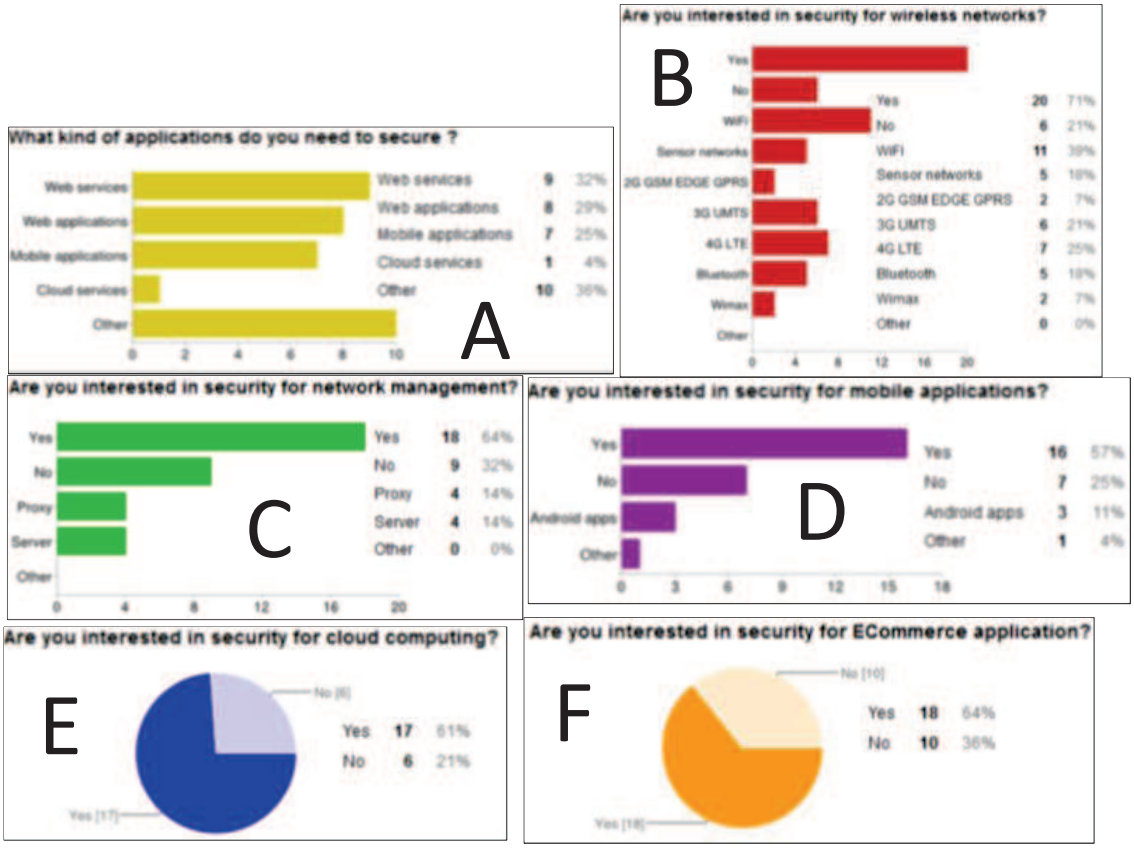

Figure 5.17: Evaluation form results from end users to update STAC with new technologies

- Are you interested in security for network management? (see Figure 5.17.C). According to the result, users are curious about security for network management. We are enriching STAC with this new domain, the related security vulnerabilities and security mechanisms. For isntance, we added popular security tools such as Ettercap, Nmap, Wireshark, Nessus, Metasploit.

- Are you interested in security for mobile applications? (see Figure 5.17.D). According to the result, users are curious about security for mobile phones. We intent to enrich STAC with this new domain, the related security vulnerabilities and security mechanisms.

- Are you interested in security for cloud computing? (see Figure 5.17.E). According to the result, users are curious about security for cloud computing. We intent to enrich STAC with this new domain, the related security vulnerabilities and security mechanisms.

- Are you interested in security for ECommerce application? (see Figure 5.17.F). According to the result, users are curious about this security domain. We intent to enrich STAC with this new domain, the related security vulnerabilities and security mechanisms.

- Programming languages used? (see Figure 5.18.A). According to the results, the most popular programming languages are Java, Javascript, PHP and Python. Thanks to these results, we are enriching the STAC database with more security mechanisms such as security framework relevant for such programming languages. For instance, for the Java language, we added popular security APIs called Java Cryptography Extension (JCE) and Java Authentication and Authorization Service (JAAS).

- Frameworks used? (see Figure 5.18.B). According to the results, the most popular frameworks are Hibernate and Spring. Thanks to these results, we are enriching the STAC database with more security mechanisms such as security framework. For instance, for the Spring framework, we added the Spring Security framework in the STAC dataset.

- Databases used? (see Figure 5.18.C). According to the results, the most popular databases uses are MySQL and PostgreSQL. Thanks to these results, we are enriching the STAC database with more security mechanisms and known vulnerabilities (e.g., SQL injection).

5.5.3 Discussions

From our point our view, the STAC knowledge base follows the semantic web best practices, which is a main added value compared to existing security ontologies. By referencing our ontologies to the LOV project, we learnt a lot and discover numerous semantic web tools to check syntaxes, best practices, etc. Regarding methodologies, it was difficult for us to decide between concepts or instances for some security mechanisms. To the best of our knowledge, we did not find any methodologies explaining this. For instance, at the beginning antivirus was an instance in the STAC dataset, and became a concept in the ontology since we added antivirus specific tools such as Norton in the dataset. The STAC knowledge base validates 'Hypothesis 8: The security knowledge base is built using the same methodology that for the M3 interoperable domain knowledge.'. Moreover, the STAC knowledge base partially validates 'Hypothesis 9: A security knowledge base can help non-experts in security to choose security mechanisms fitting their needs to secure IoT applications'. The concepts of security are not so intuitive for non experts, there is a real need to vulgarize as much as possible security for non-experts. We have to improve the user interface that we have proposed and the explanations to assist users in securing their applications. STAC is a first step in this research problem, and requires to investigate much more how we can assist users in securing their applications. STAC could be improved by hiding security to users by automatically choosing the security mechanism fitting the need of the applications and even by setting up the security mechanisms. According to the evaluation results, we are extending the STAC knowledge base with new technologies as recommended by the users

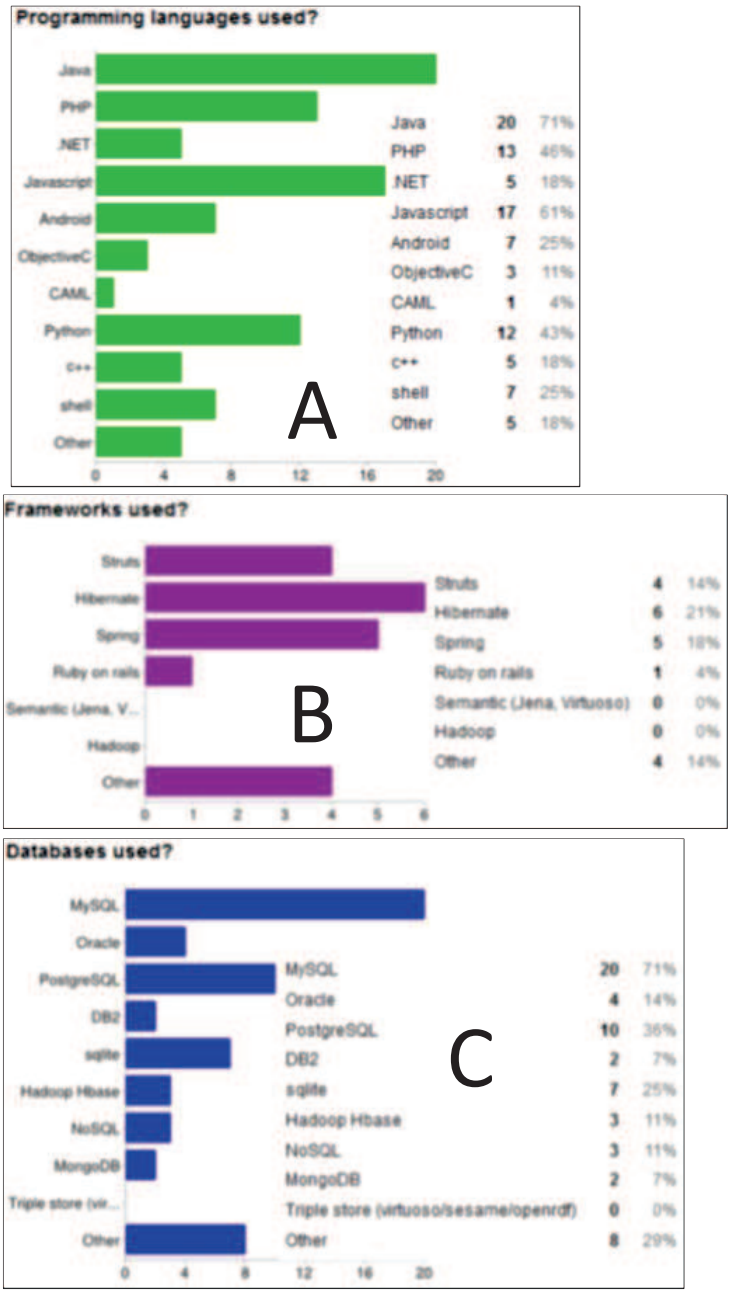

Figure 5.18: Evaluation form results from end users to update STAC with web technologies

such as wireless networks, network management, mobile application, cloud, e-commerce, web, etc.

5.6 The novelty of the STAC knowledge base

In section 2.4.4, we explained that existing security ontologies had shortcomings. These shortcomings are overcome as following.

- STAC has been designed to secure IoT and M2M, more precisely to assist users in securing applications or architectures.
- Lack of unify terms has been solved thanks to the STAC nomenclature, an essential step to easily combine security domains.
- STAC is a cross-domain security knowledge base which covers numerous security technologies: sensor networks, Wi-Fi, 2G, 3G, 4G, web applications, network management. The going work is ocerving new technologies such as Wimax, Zigbee, Blutooth, Mesh networks, and Manet.
- STAC is a more complete knowledge base since we: (1) classified both threats and security mechanisms according to various technologies, (2) classified attacks and security mechanisms according to the OSI model, (3) described strengths and weaknesses of security mechanisms, and (4) specified the relationships between security mechanisms, attacks and security properties.
- Follow semantic web best practice: (1) to reuse the domain knowledge, (2) be referenced on the LOV catalogue, (3) check and fixes syntaxes detected by validator tools, and (4) follow recommendations from semantic web experts or tools to improve our knowledge base.
- STAC generator reused the STAC knowledge base which is the main novelty of this work since it assists users in securing applications.

5.7 Concluding Remarks

In this chapter, we have presented the STAC cross-domain security knowledge base to assist developers and designers who are not expert in security to secure their IoT/M2M applications or architectures. However, STAC can be used to secure any kind of applications. STAC has been designed using the same approach described in the previous chapters (Chapter 3 and Chapter 4). The STAC generator is inspired from the SWoT generator and the STAC cross-domain knowledge from the M3 interoperable domain knowledge. STAC classifies numerous technologies and their attacks, the existing security mechanisms, security properties, features, etc. STAC covers various security domains: sensor networks, cellular networks (2G, 3G, 4G), wireless networks (sensor networks, Wi-Fi, Bluetooth, RFID,

Wimax, Zigbee, Manet, Mesh), web applications and network management. The STAC ontology and the dataset follow the semantic web best practices, are published online and have been referenced by the Linked Open Vocabularies catalogue. Further, the STAC generator exploits this security knowledge base to assist users in securing applications. The STAC security knowledge base has been employed to build a cross-domain security application that we describe in section 6.6.

As future work, we intent to automatically enrich the STAC security knowledge base with new technologies, attacks and security mechanisms by improving the user-friendly interface and make it more user-friendly. Another step will be to automatically integrate in the application to secure the security mechanisms (e.g., encrypting data with cryptographic algorithms) by generating the code required. Further taking inspiration from S-LOR, we could integrate STAC rules such as 'under attacks' to check the robustness or vulnerability of the application.

In the next chapter (Chapter 6), we show through five uses cases how M3 or STAC can be used.

Part III

Use Cases & Conclusions

In this last part, we will first present in Chapter 6, the uses cases of the M3 framework and in Chapter 7, we will conclude the thesis and outline future work.

Chapter 6

M3 Framework at Work

"Knowing is not enough; we must apply. Willing is not enough; we must do."

Bruce Lee

In this chapter, more precisely, in section 6.1, we first define the different stakeholders who can use or enrich the M3 framework. Then, in the next sections, we focus on five use cases where the M3 framework has been and can be applied by developers or can be employed by end-users. More precisely, in section 6.2, we explain how M3 has been employed by developers to build cross-domain mobile SWoT applications. In section 6.3, we demonstrate how such applications are used in a car dashboard. In section 6.4, we present how end-users can use M3 applications embedded in smart fridges. In section 6.5, we present how end-users can use M3 applications embedded in smart luggage. Finally, in section 6.6, we describe how the Security Toolbox: Attacks and Countermeasures (STAC) component presented in Chapter 5 can help developers. As said in Chapter 5, STAC assists developers in securing IoT architectures or applications by suggesting security mechanisms fitting their needs. Finally, we conclude this chapter in section 6.7.

6.1 Using and Contributing to M3

In Figure 6.1, we show the different stakeholders of the M3 framework: software developers, domain experts, semantic web experts, commercial gateways, commercial devices and standardizations experts.

Software developers get high level abstractions from sensor data and cross-domain suggestions provided by the M3 framework. The developers have the choice between visualizing these results in a user-friendly interface, sending alerts or controlling actuators. In section 3.4, we explained that developers can use the M3 web services thanks to the M3 API documentation[1] to: (1) look for M3 templates, (2) download the M3 templates, and (3) semantically annotate sensor data. Such web services can be employed in the cloud (explained in section 3.6) but also in mobile devices (see section 6.2) or even in gateways. To help developers in designing IoT applications, we have created M3 templates which can be

[1] `http://www.sensormeasurement.appspot.com/?p=documentation`

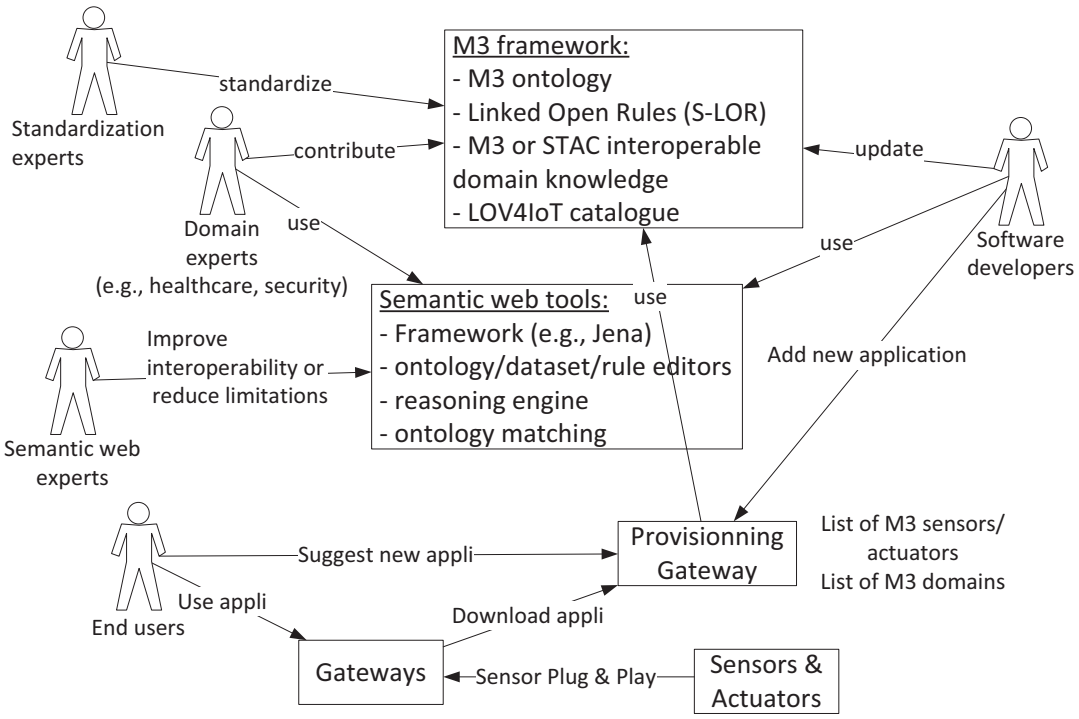

Figure 6.1: Different stakeholders could be involved in the M3 framework

used to build the SWoT applications without learning semantic web applications (explained in 3.4).

A second scenario is explicated in section 6.6 to assist developers in securing their applications. Once, the developers indicate the technology employed in their application, the STAC application displays the corresponding attacks and security mechanisms.

Domain experts can contribute to the 'Linked Open Rules' and LOV4IoT by sharing their knowledge expertise: ontologies, datasets and rules. With the help of semantic web experts, the domain experts can contribute to the M3 or the STAC interoperable domain knowledge bases. For instance, healthcare experts can contribute to the M3 healthcare knowledge base by integrating new rules to interpret health measurements. Security experts can contribute to the STAC knowledge base by integrating new security domains such as security for Bluetooth or RFID and the related attacks and countermeasures. Domain expert researchers can also exploit the LOV4IoT web page to compare their works to the state of the art.

Semantic web experts can contribute to the M3 framework by working on the interoperability issues between ontology and rule editor tools. They could encourage best practices through their tools too. Ontology matching tool experts could evaluate their tool with the new LOV4IoT dataset, instead of the popular benchmark Ontology Alignment

Evaluation Initiative (OAEI)[2] used to evaluate ontology matching tools. Evaluating future ontology matchings tools with these two datasets will demonstrate the maturity of ontology matching tools since they will be relevant for all ontologies developed with ontology softwares and by non-semantic web experts. For building the LOV4IoT dataset, we collected a great deal of domain ontologies. Further, we encouraged domain experts to share their domain knowledge on the web and to follow semantic web best practices. Thanks to this tremendous work, the Linked Open Vocabulary (LOV)[3] claim us as contributors of their community.

Commercial gateways could integrate our M3 framework directly inside the gateway to sell a smart gateway. The gateway just needs a first connection to the Internet for the set up phase to download the required M3 template. These gateways could be embedded in a smart car, smart fridge or smart home. An example of our smart car scenario could be embedded in car's dashboard (see section 6.3). The naturopathy scenario could be embedded in the smart kitchen or the smart refrigerator (see section 6.4). The tourism scenario is relevant if it is embedded in a luggage to suggest garments according to your destination and the weather forecasting (see section 6.5).

End users enjoy the applications generated by M3. These IoT applications could be accessible in Google Play, App store, etc. They can also contribute to the M3 framework by suggesting new IoT applications that we can easily integrate as a new template. An example is explained in section 6.3, 6.4 and 6.5.

Commercial devices could follow our M3 nomenclature to describe sensor measurement to ease the automation of IoT/M2M to make data interoperable and facilitate them interpretation.

Standardization experts could use our M3 framework as a basis for further standardizations. The M3 nomenclature and M3 ontology provide interoperability to easily interpret sensor data. They are relevant for standardization bodies like oneM2M, ETSI M2M, W3C Web of Things and W3C SSN ontology. The uniform descriptions have already been communicated to oneM2M WG-5 (MAS)[4] [Gyrard and Bonnet, 2014]. We referenced all semantic bad practices and suggested the related guidelines in a draft document [Gyrard and Bonnet, 2014] to the oneM2M international standard. We explained to the W3C Web of Things Interest Group the interoperability issues for combining domain knowledge to build cross-domain applications [Gyrard et al., 2014a]. W3C SSN ontology could standardize our M3 ontology and S-LOR to provide a basis for reasoning on sensor data as explained in [Gyrard et al., 2014]. We designed an M3 interoperable domain knowledge which could be used as drafts to standardize domain ontologies relevant for IoT.

6.2 Developing Mobile SWoT Applications with M3

The same scenarios presented in section 3.6 have been developed on Android-based devices and show that our M3 approach is feasible and flexible to integrate semantics in constrained

[2] `http://oaei.ontologymatching.org/`
[3] `http://lov.okfn.org/dataset/lov/`
[4] http://onem2m.org/MAShome.cfm

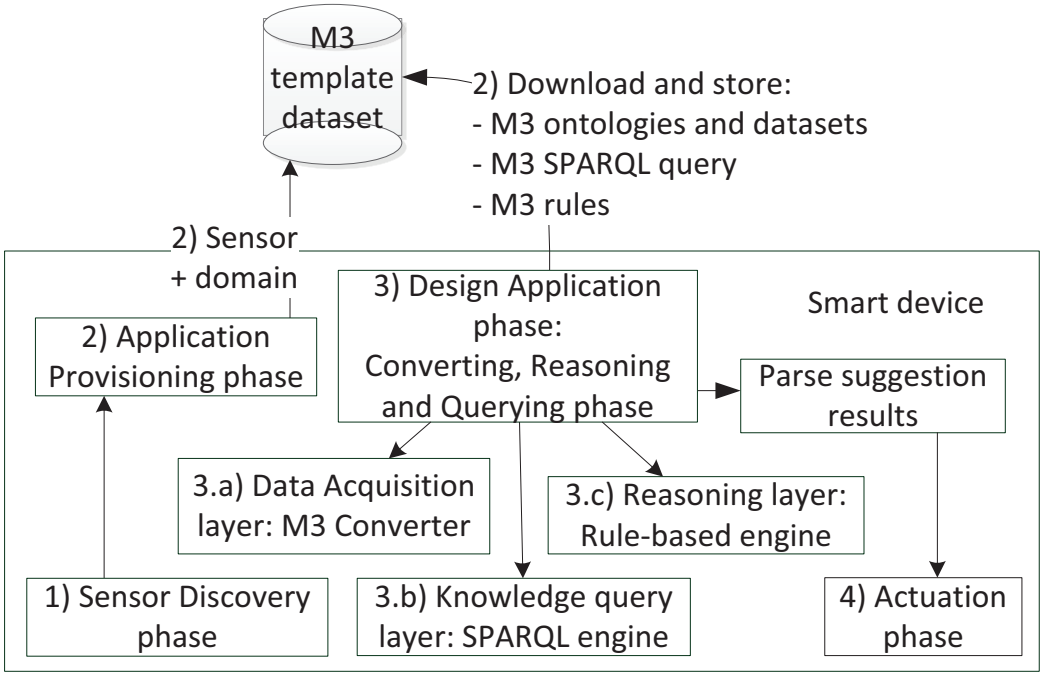

Figure 6.2: M3 architecture for mobile devices

devices such as mobile devices (e.g., smart phones, tablets). In this section, we explain how this could be done through four steps (see Figure 6.2):

- The **sensor discovery phase** detects sensors employed. This work is not explained in this thesis, since it is based on the work designed by Datta et al. which is in progress [Datta, 2015]. Based on their work, we know the sensors recognized by the systems.

- The **application provisioning phase** enables the smart device asking through Internet the M3 template dataset to automatically download the knowledge required to build the IoT applications for sensors and domains given by the sensor discovery phase. The M3 template dataset returns a SWoT template with all the domain knowledge needed to later interpret sensor data and combine it with cross-domain domain knowledge.

- The **design application phase** uses the M3 converter and M3 rules embedded in the M3 template to convert sensor data in a unified way. The sensor data comes from the smart device which gets sensor data encoded according to the SenML format provided by the gateway. Secondly, a reasoning engine interprets sensor data and infer additional knowledge. Thirdly, the sensor data inferred is linked to cross-domain datasets provided in the M3 template to obtain cross-domain suggestions.

- The **actuation phase** can be performed according to the cross-domain suggestions provided by M3. This work is not explained in this thesis, since it is based on the work designed by Datta et al. which is in progress [Datta, 2015].

Instead of using Jena, the lightweight version AndroJena has been used in Android-powered devices for technical reasons. AndroJena includes the reasoning engine and the SPARQL engine. The Android-powered devices just need a first connection to Internet to download the Semantic Web of Things (SWoT) template. The template contains the domain knowledge required to interpret sensor data to provide cross-domain suggestions. The M3 processing is done locally to semantically annotate, reason, enrich and query sensor measurements. Let us now see in more details how these steps can be performed (see Figure 6.2).

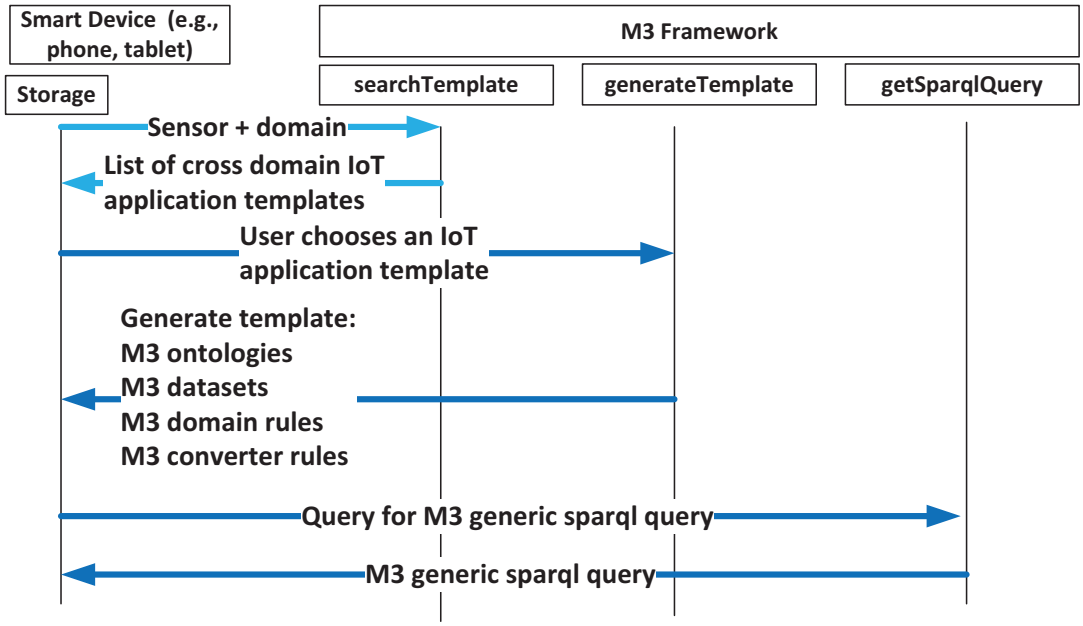

Figure 6.3: Sequence diagram of application provisioning phase

6.2.1 Application Provisioning Phase

Once the sensor discovery phase is achieved, the user selects a sensor (e.g. light sensor) and an associated domain (e.g. weather). The mobile application then queries a web service from the M3 framework with the sensor and domain information. The M3 framework internally retrieves a list of previously defined cross-domain scenarios involving the selected sensor and domain. The application receives and presents the list to the user. For example, based on light sensor and weather domain, the M3 framework will propose four cross-domain scenarios: (i) Weather, Luminosity and Emotion, (ii) Weather, Tourism and Clothes, (iii)

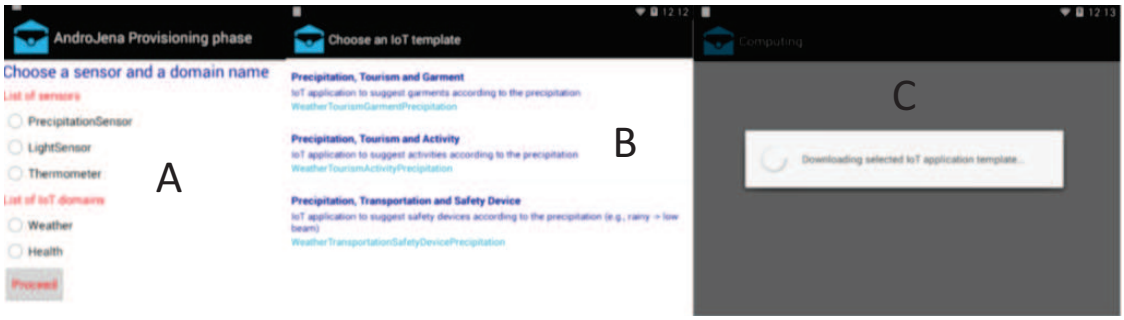

Figure 6.4: Application provisioning phase designed on the Android-powered mobile phone

Weather, Tourism and Activities and (iv) Weather, Transportation and Safety Device. Each scenario accomplishes a different goal for the user. The 'Weather, Tourism and Activities' scenario is useful when a user is in vacation as the application in this case will propose activities based on outside weather. Depending on the requirement, the user selects one scenario and the mobile application queries another web service in the M3 framework to download the related application template. Each application template contains the M3 ontologies for the domains, M3 datasets, M3 rules and a generic SPARQL query. The above steps of the provisioning phase are illustrated in Figure 6.3. The downloaded template contains all the domain knowledge necessary to interpret sensor data and combine the inferred data with cross-domain knowledge from the weather and tourism domains. The proof of concept on Android-power mobile phone is displayed in Figure 6.4.

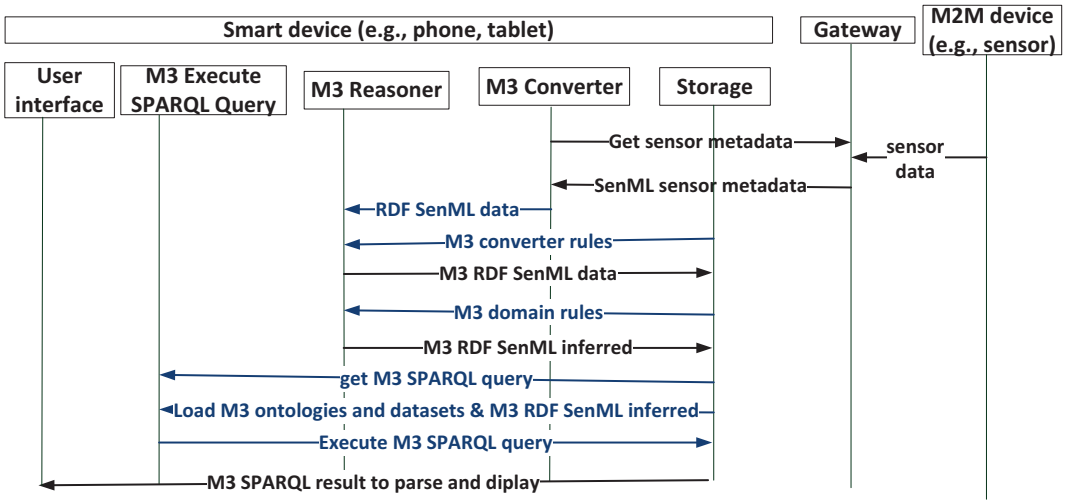

Figure 6.5: Sequence diagram of the design application phase

6.2.2 Design Application Phase

After the application provisioning phase, the M3 converter queries the gateway to get the SenML sensor metadata. The M3 converter then semantically annotates these metadata with Resource Description Framework (RDF) using "M3 converter rules" so that converted RDF sensor data is expressed in an uniform way. This step is necessary as the real sensor data can come from any gateway which may or may not follow an uniform nomenclature. In case the M3 nomenclature is followed, this step can be avoided.

To infer additional knowledge, the mobile application uses the M3 reasoner again with "M3 domain rules". This generates a high level abstraction (called deduction) from the sensor measurement type, measurement value and associated domain name. The inferred data is updated in the storage. Finally, the application loads the M3 cross-domain knowledge and inferred data to execute the SPARQL query which gives one or more cross-domain "suggestions". These deduction(s) and suggestion(s) are reported to the user through an user interface or a notification bar. These above steps are illustrated in Figure 6.5 and related to the functionalities of the reasoning layer, knowledge query layer and application layer of the M3 framework.

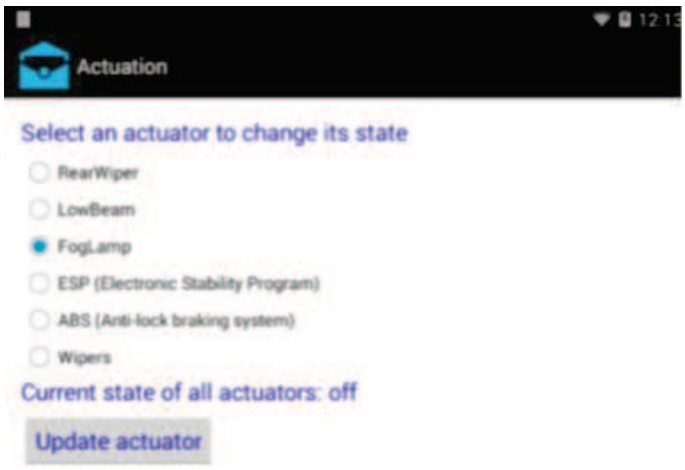

Figure 6.6: Actuation phase designed on the Android-powered mobile phones

Finally, in the actuation phase, mobile developers will parse M3 suggestions and decide if notifications are sent to the users or control actuators (e.g., switch on/off) as displayed in Figure 6.6.

6.3 Integrating M3 in Smart Cars

In the previous section, we have seen how mobile developers can easily use the M3 framework in Android-based constrained devices. An application of this work enables displaying the

results returned by M3 in a user-friendly interface embedded in the car dashboard. This application could provide suggestions to the driver according to the weather conditions and driver's state as depicted in Figure 6.7.

Figure 6.7: M3 integrated in car dashboard

Such applications are possible thanks to the sensor discovery phase where the precipitation sensor embedded in a smart car is automatically recognized. Afterwards, M3 can easily retrieve a M3 template (see Figure 6.4) involving this sensor, this is the application provisioning phase. Figure 6.7 enables providing suggestions provided by the M3 reasoning and querying which is possible because the M3 template has been downloaded previously. The result of this application is displayed in a user-friendly interface which suggests to switch on fog lights because M3 interprets that it is rainy and provides such suggestions. Finally, the actuation phase will be done as displayed in Figure 6.6.

Figure 6.7 is a mock-up to show a user-friendly interface for end users. The real demonstration which has been implemented on the cloud has been presented in section 3.6, "Scenario 1: Suggesting safety devices according to the weather". The transportation application can be tested online[5] and is comprised of two sub-applications:

[5] `http://www.sensormeasurement.appspot.com/?p=transport`

- Suggesting safety devices in the smart car according to the weather.
- Suggesting safety devices in the smart car when it is snowy.

To build such applications, we simulated sensor datasets since we did not have the opportunity to exploit real sensors. Such datasets are accessible online too. The weather dataset[6] simulates luminosity, temperature, wind speed, humidity and precipitation measurements. The snow dataset[7] simulates only two measurements: precipitation and temperature. This dataset is mainly used to apply more complicated rules which involve two measurements at the same time.

The transportation knowledge base has been redesigned manually and has been inspired by the domain knowledge that we referenced, classified and synthesized in the LOV4IoT dataset. More precisely, it has been inspired by the following works:

- Autonomous vehicle assistance [Morignot and Nashashibi, 2012] [Pollard et al., 2013]. The authors describe relationships between weather, road conditions and driver's state.
- Context-aware driver assistance systems [Fuchs et al., 2008b] [Fuchs et al., 2008a].
- Road traffic management [Bermejo et al., 2013].
- Driver fatigue detection [Deshmukh et al., 2011].
- Rules to deduce weather state [Staroch, 2013]. This work has been applied to the smart home domain, but we reuse the domain knowledge in another context, the transportation domain.

These works are from heterogeneous domains such as transportation, healthcare, affective sciences and weather. To design the transportation knowledge base, which is a hub combining these four domains, we reused the methodology explained in section 4.4.

The transportation knowledge base has been reused in the template exploited to build transportation applications, it is comprised of: (1) the rules to semantically annotate sensor data according to the M3 nomenclature, (2) the transportation[8], M3[9] and weather[10] ontologies in OWL/XML, (3) the transportation[11] and weather datasets[12] in RDF/XML, and (4) the Linked Open Rules[13] specific to the weather domain, a dataset of rules to interpret weather measurements.

For instance, the transportation template is provided when the precipitation sensor and the weather domain are chosen on the SWoT generator user interface or with web services.

[6] `http://www.sensormeasurement.appspot.com/dataset/sensor_data/weatherData_8KB_17Septembre2014.rdf`

[7] `http://www.sensormeasurement.appspot.com/dataset/sensor_data/snow_dataset.rdf`

[8] `http://www.sensormeasurement.appspot.com/ont/m3/transport`

[9] `http://www.sensormeasurement.appspot.com/ont/m3/m3`

[10] `http://www.sensormeasurement.appspot.com/ont/m3/weather`

[11] `http://www.sensormeasurement.appspot.com/dataset/transport-dataset`

[12] `http://www.sensormeasurement.appspot.com/dataset/weather-dataset`

[13] `http://www.sensormeasurement.appspot.com/RULES/LinkedOpenRulesWeather.txt`

This template called 'Precipitation, Transportation and Safety Devices' will suggest safety devices according to the weather measurements, more precisely, when sunny, cloudy or rainy are detected. Other templates are provided when luminosity, cloud cover, wind speed, etc. sensors are selected.

6.4 End-User Centric Approach: M3 Embedded in Smart Fridges

Figure 6.8 shows that M3 could be integrated in smart fridges. For instance, users take their body temperature with a sensor connected to the smart fridge. The temperature measured is 40 degree Celsius which is automatically interpreted by the smart fridge which deduces that the user has fever. In the background, the Sensor-based Linked Open Rules (S-LOR) is running, more precisely the reasoning engine and the M3 rules provided in the naturopathy template that has been embedded in the smart fridge. Then, since the naturopathy template has been integrated in the fridge, food datasets and their relationships with health datasets are available. In the downloaded datasets, they are relationships between home remedies and symptoms such as fever. Finally, the fridge suggests some home remedies such as honey, lemon or thyme tea to help users fight microbes. The users can trust this information, since the fridge even provides information about such remedies [Sharma et al., 2012].

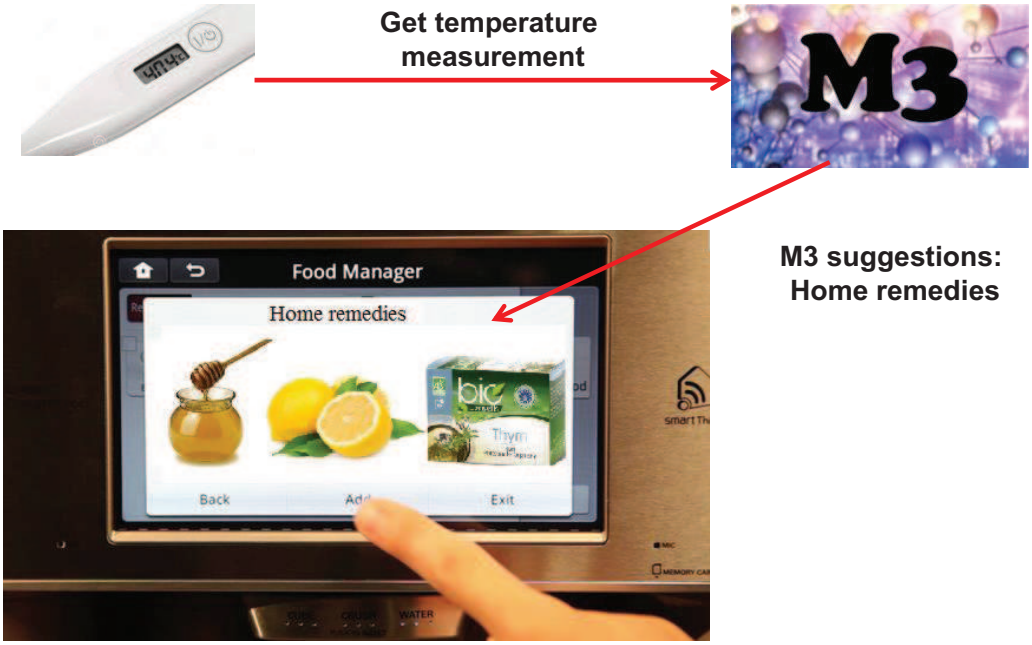

Figure 6.8: M3 embedded in smart fridges to suggest food or home remedies

Figure 6.8 is a mock-up to show a user-friendly interface for end users. The real demonstration which has been implemented on the cloud has been presented in section 3.6, "Scenario 3: Suggesting home remedies according to health measurements". The naturopathy application can be tested online[14] and is comprised of five sub-applications:

- Suggesting home remedies according to the body temperature.
- Suggesting food according to the outside temperature.
- Deducing mood according to the external luminosity.
- Deducing mood or diseases from heart beat, skin conductance and blood pressure.
- Suggesting a recipe according to the food available in your kitchen.

To build such applications, we simulated sensor datasets since we did not have the opportunity to exploit real sensors. Such datasets are accessible online too. The health dataset[15] simulates heart beat, temperature, blood pressure, cholesterol and skin conductance measurements. The weather dataset[16] simulates luminosity, temperature, wind speed, humidity and precipitation measurements.

The naturopathy knowledge base has been redesigned manually and has been inspired by the domain knowledge that we referenced, classified and synthesized in the LOV4IoT dataset. More precisely, it has been inspired by the following works:

- RFID embedded on food [Gu and Wang, 2009] [Xie et al., 2013] and food traceability [Pizzuti and Mirabelli, 2013].
- Food recommender systems [Fudholi et al., 2009] [Suksom et al., 2010] or recipe recommendation systems [Freyne et al., 2011] [Erschbamer and Malleier, 2012] [Kalem and Turhan, 2005] such as WikiTaaable [Blansché et al., 2010], dietary recommendations [Su et al., 2012] and dietary recommendations for athletes [Tumnark et al., 2013].
- Affective sciences describing emotions [Hastings et al., 2011] [López et al., 2008], effects of weather parameters on mood [Denissen et al., 2008] or even relationships between emotion and color [Nijdam, 2009]. ColorCocktail is an ontology-based recommender system suggesting cocktails according to the mood and user preferences [Chen et al., 2006].
- Relationships between diseases and environmental causes (stress, family conditions, drugs, climate, pollution, noise) or even weather and mood [Hadzic et al., 2008]. For instance, people are happy when it is sunny. Another example is that a lack of vitamin B leads to depression. In [Truong et al., 2011], the authors explain the relationships between diseases (e.g., osteoarthritis) and the climate (e.g., rainy).
- IoT-based sportsman/ woman monitoring application [Rodríguez-Molina et al., 2013].

[14] http://www.sensormeasurement.appspot.com/?p=naturopathy

[15] http://www.sensormeasurement.appspot.com/dataset/sensor_data/senml_m3_health_data.rdf

[16] http://www.sensormeasurement.appspot.com/dataset/sensor_data/weatherData_8KB_17Septembre2014.rdf

- Relationships between food and diseases such as diabetes [Cantais et al., 2005].
- Relationships between food and recipes with the SmartProducts project[17].
- Relationships between food and climate [Miao et al., 2013].
- Rules to deduce weather state [Staroch, 2013]. This work has been applied to the smart home domain, but we reuse the domain knowledge in another context, the transportation domain.

These works are from heterogeneous domains such as healthcare, food, weather and affective sciences. To design the naturopathy knowledge base, which is a hub combining these four domains, we reused the methodology explained in section 4.4.

The naturopathy knowledge base has been reused in the template exploited to build naturopathy applications, it is comprised of: (1) the rules to semantically annotate sensor data according to the M3 nomenclature, (2) the naturopathy[18], M3[19] and health[20] ontologies in OWL/XML, (3) the naturopathy[21] and health datasets[22] in RDF/XML, and (4) the Linked Open Rules[23] specific to the health domain, a dataset of rules to interpret health measurements.

For instance, the naturopathy template is provided when the thermometer sensor and the healthcare domain are chosen on the SWoT generator user interface or with web services. This template called 'Body temperature, symptoms and home remedies' will suggest home remedies according to the body temperature, more precisely, when the fever is detected. Another example of template can provide the naturopathy knowledge base when the thermometer sensor and the weather domain are chosen on the SWoT generator user interface or with web services. This template called 'Outside temperature, Season and Food' will suggest food according to the season, more precisely, when the season is related to the outside temperature.

6.5 End-User Centric Approach: M3 Embedded in Smart Luggage

Figure 6.9 shows that M3 could be integrated in smart luggage. For instance, Nelly loves to decide on the last minute her vacation to benefit from reductions. She goes to the LastMinute.com web site to find her destination. She found an excellent destination with a big reduction. She has to leave in 2 hours, M3 will aid her to pack their clothes in her smart luggage. M3 is aware of the weather forecasting in her destination, and will suggest

[17] http://projects.kmi.open.ac.uk/smartproducts/
[18] http://www.sensormeasurement.appspot.com/ont/m3/naturopathy
[19] http://www.sensormeasurement.appspot.com/ont/m3/m3
[20] http://www.sensormeasurement.appspot.com/ont/m3/health
[21] http://www.sensormeasurement.appspot.com/dataset/naturopathy-dataset
[22] http://www.sensormeasurement.appspot.com/dataset/health-dataset
[23] http://www.sensormeasurement.appspot.com/RULES/LinkedOpenRulesHealth.txt

her appropriate clothes for her vacation. For instance, to go in an exotic country, she will bring swimsuit, sunglasses, hat, etc.

In the same time, Guillaume also uses the LastMinute.com to go to the mountains in winter. M3 will suggest him the required clothes and equipments such as glove, scarf, turtleneck, pull, etc.

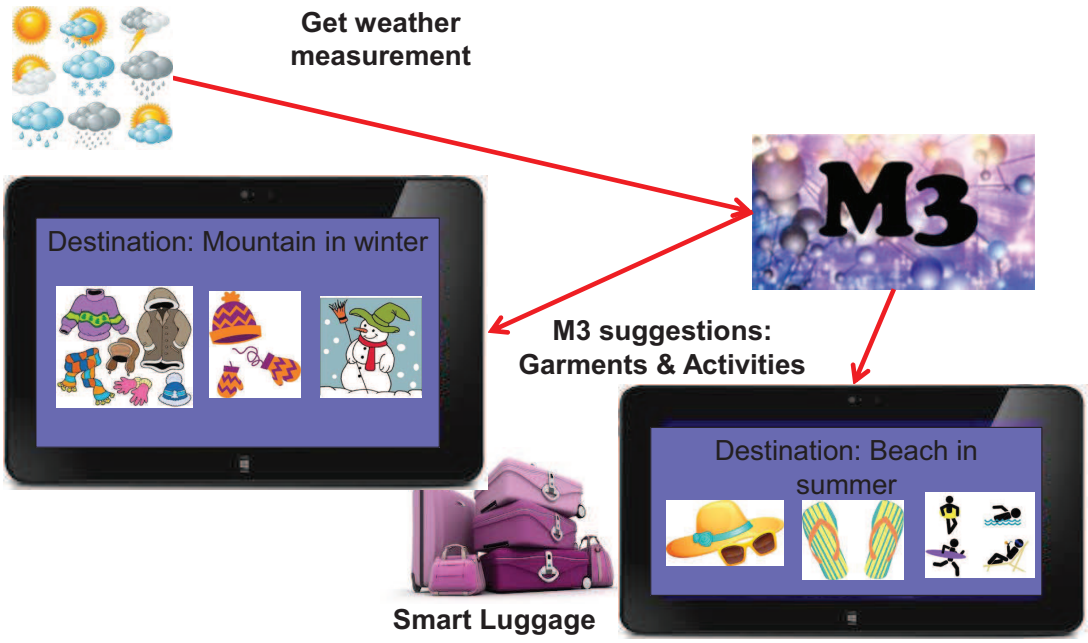

Figure 6.9: M3 embedded in smart luggage to suggest garments and activities

Figure 6.9 is a mock-up to show a user-friendly interface for end users. The real demonstration which has been implemented on the cloud has been presented in section 3.6, "Scenario 2: Suggesting activities or clothes according to the weather". The tourism application can be tested online[24] and is comprised of four sub-applications:

- Suggesting activities according to the weather.
- Suggesting clothes according to the weather.
- Suggesting garments when it is snowy.
- Suggesting activities when it is snowy.

To build such applications, we simulated sensor datasets since we did not have the opportunity to exploit real sensors. Such datasets are accessible online too. The weather dataset[25]

[24] http://www.sensormeasurement.appspot.com/?p=tourism

[25] http://www.sensormeasurement.appspot.com/dataset/sensor_data/weatherData_8KB_17Septembre2014.rdf

simulates luminosity, temperature, wind speed, humidity and precipitation measurements. The snow dataset[26] simulates only two measurements: precipitation and temperature. This dataset is mainly used to apply more complicated rules which involve two measurements at the same time.

The tourism knowledge base has been redesigned manually and has been inspired by the domain knowledge that we referenced, classified and synthesized in the LOV4IoT dataset. More precisely, it has been inspired by the following works:

- Suggesting activities according to the weather or season (e.g., museum if it is rainy) [Codina et al., 2013] [Codina and Ceccaroni, 2010c] [García-Crespo et al., 2011]. Further, the authors mentioned the idea of a domain-independent recommender system.
- Classification of sports, leisure and hobbies [Blanco-Fernández et al., 2011].
- The CHIP demonstrator, an art recommender [Wang et al., 2010].
- Relationships between accommodation, attraction and cultural event [Kuntarto and Gunawan, 20
- SWRL-based context reasoning to deduce the tourist type (e.g., bar lover or art lover) [Ziafati et al., 2011].
- Personalized-travel companion [Figueiras et al., 2013] [Dema, 2008]. In [Figueiras et al., 2013], the authors combine data from several datasets such as GooGle Places, Google Calendar, TripIt, Foursquare, OpenStreetMap, openCyc and Facebook. In [Dema, 2008], the authors classify events, attractions and accommodations.
- The travel and user ontologies used to answer such questions: 'Today I want to do some sightseeing in Shanghai and then go to sea, can you give me some suggestions?' [Wang et al., 2008]. The authors classified kind of food, accommodation, transportation, shopping and attractions.
- Destination Context Ontology (DCO) to describe relationships between weather temperature, scenery, volume of traffic, crime rate and status of the destination [Daramola et al., 2009]
- Description and classification of tourist offer, accommodation rating, transport services, activities and interests [Damljanović and Devedzic, 2008] [Damljanovic and Devedzic, 2008]
- Rules to deduce weather state [Staroch, 2013]. This work has been applied to the smart home domain, but we reuse the domain knowledge in another context, the transportation domain.

These works are from heterogeneous domains such as tourism and weather. To design the tourism knowledge base, which is a hub combining these two domains, we reused the methodology explained in section 4.4.

The tourism knowledge base has been reused in the template exploited to build tourism applications, it is comprised of: (1) the rules to semantically annotate sensor data according

[26] `http://www.sensormeasurement.appspot.com/dataset/sensor_data/snow_dataset.rdf`

to the M3 nomenclature, (2) the tourism[27], M3[28] and weather[29] ontologies in OWL/XML, (3) the tourism[30] and weather datasets[31] in RDF/XML, and (4) the Linked Open Rules[32] specific to the weather domain, a dataset of rules to interpret weather measurements.

For instance, the tourism template is provided when the thermometer sensor and the weather domain are chosen on the SWoT generator user interface or with web services. This template called 'Temperature, Tourism and Garment' will suggest garments according to the weather measurements, more precisely, when sunny, cloudy or rainy are detected. Another template is provided with the same sensor and domains selected: 'Temperature, Tourism and Activity'. Other templates are provided when precipitation, cloud cover, wind speed, etc. sensors are selected.

6.6 Designing Secure IoT Applications with STAC

In Chapter 5, we explained that the purpose of the STAC ontology and dataset is to aid developers and project managers to secure IoT-based applications. In this section, we will see how STAC can be used by developers. STAC provides an user interface that allows developers to query the STAC knowledge base, though several web services that we developed, to obtain the required security information to secure their IoT applications as displayed in the menu tab (Figure 6.10). The 'Security 'tab is composed of sub-tabs which covers all security aspects of this thesis. The developers who want to secure their applications can access to:

- The STAC application user interface (see Figure 6.11). First, the developers choose a specific technology. Then, STAC displays all related attacks and security mechanisms. For each security mechanism, STAC indicates the properties satisfied (e.g., authentication) and the advantages and shortcomings (e.g., secured or deprecated). Figure 6.11 illustrates through an example how the developers can use STAC. In this figure, we can see that, the developers choose a Wi-Fi technology. Then, all attacks displayed (e.g., Steal NIC) and security mechanisms specific to the WiFi technology are displayed. After that, the developers choose a security mechanism which is in this example WPA2. Finally, the user interface provides the advantage of this mechanism which is 'secured' and the security property satisfied which is 'authentication'. For each drop-down list, a tooltip is displayed to provide definitions of each suggestion (attack, technology, security mechanism).

- Frequently Asked Questions (FAQ) related to the security domain.

- Network management

[27] `http://www.sensormeasurement.appspot.com/ont/m3/tourism`
[28] `http://www.sensormeasurement.appspot.com/ont/m3/m3`
[29] `http://www.sensormeasurement.appspot.com/ont/m3/weather`
[30] `http://www.sensormeasurement.appspot.com/dataset/tourism-dataset`
[31] `http://www.sensormeasurement.appspot.com/dataset/weather-dataset`
[32] `http://www.sensormeasurement.appspot.com/RULES/LinkedOpenRulesWeather.txt`

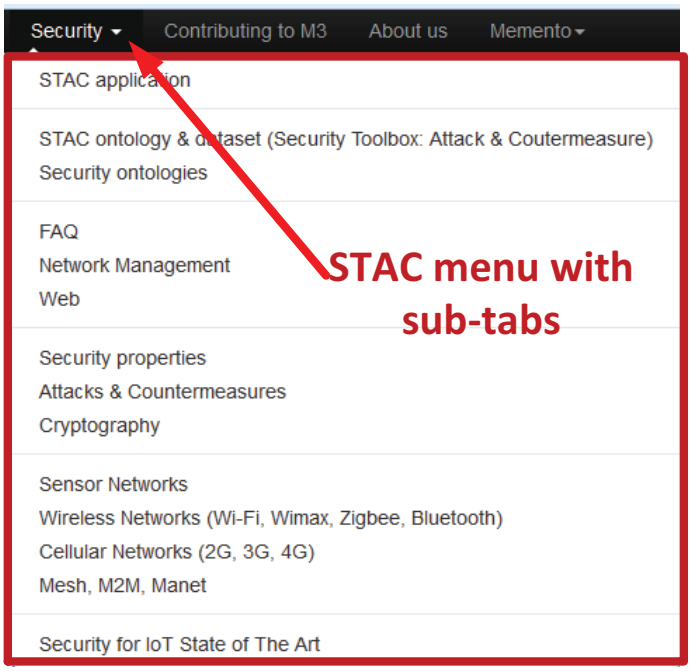

Figure 6.10: STAC menu with sub-tabs

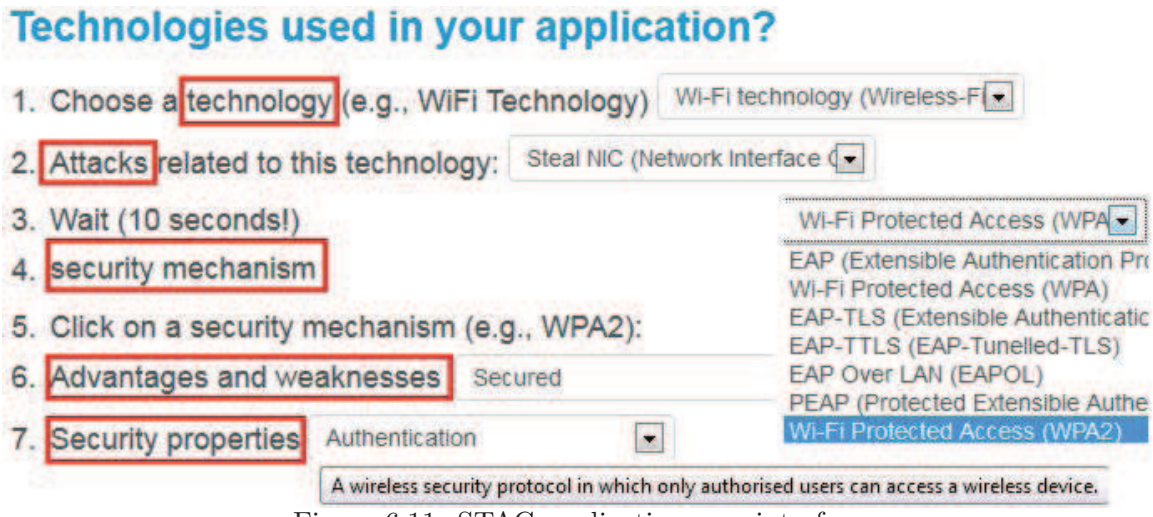

Figure 6.11: STAC application user interface

- Web
- Security properties (see Figure 6.12).

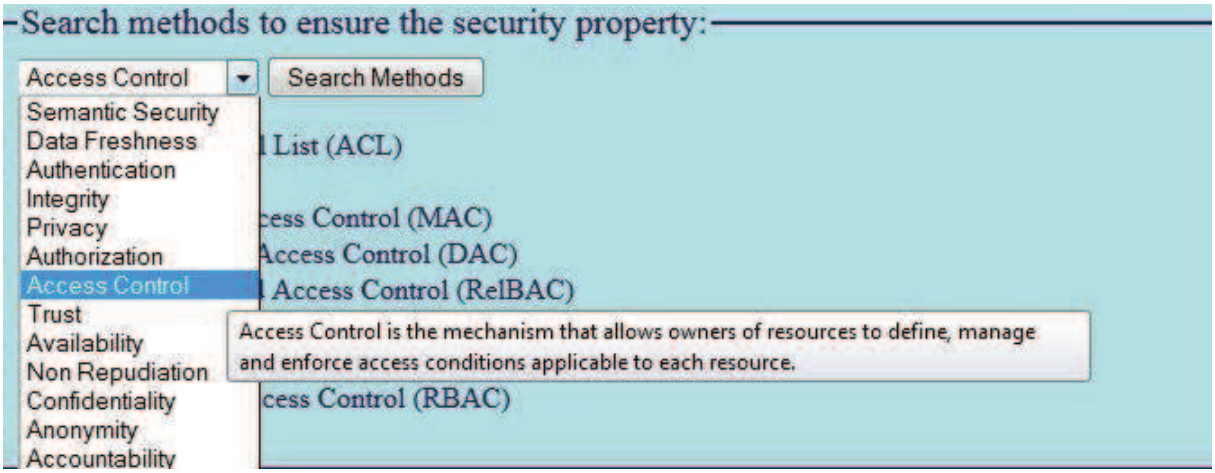

Figure 6.12: Security properties user interface

- Attacks and countermeasures (see section 5.4, Figure 5.13)
- Cryptography (see section 5.4, Figure 5.11)
- Sensor networks. The software developers who need information about security in sensor networks go through the menu security tab to the user interface[33] specific to sensor networks (see Figure 6.13). The sensor network user interface gives information about sensor protocols, sensor attacks, sensor security mechanisms and sensor key managements). This interface indicates which security algorithms are used in sensor protocols (e.g., the SPINS sensor protocol uses the RC6 algorithm). It also indicates how sensor keys are managed. For example, in the case of the LEAP sensor key management; the interface lists the LEAP four keys: pairwise key, cluster key, group key and individual key. A tooltip gives more information about all concepts: the definition of threats or security mechanisms. A click on each drop-down list shows all sensor protocols, sensor attacks, etc.
- Wireless networks with Wi-Fi.
- Security for IoT state of the art. The user can look at this user interface to search specific research articles published in conferences that we have classified for all of the mentioned technologies before. These articles guide us to build the STAC cross-domain security knowledge base. Further, they have been referenced in our knowledge base to convince developers that they can trust the STAC knowledge base.

[33] `http://www.sensormeasurement.appspot.com/?p=sensor`

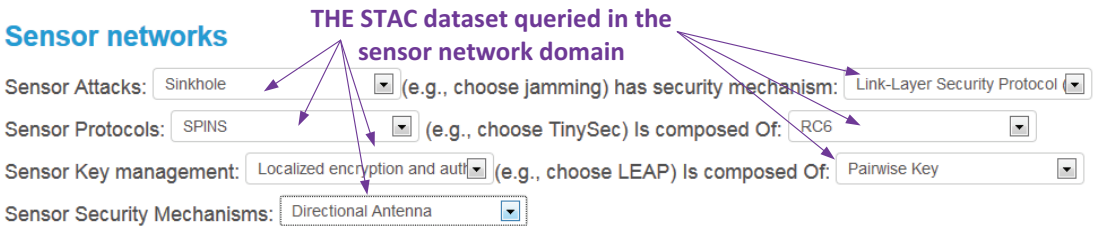

Figure 6.13: Security for sensor networks user interface

The security experts who want to design or reuse security ontologies will refer to the security tabs (see Figure 6.10):

- The STAC ontology and dataset. Security experts are redirected to the LOV4IoT dataset and will search the STAC ontology. They will find links to these files and related research articles.
- Security ontologies. The user will be redirected to the LOV4IoT user interface.

6.7 Concluding Remarks

In this chapter, we first presented the potential stakeholders of the M3 framework such as developers, end users, semantic web experts, domain experts, standardization experts and commercial devices or gateways. Then, we described five use cases. The first use case concerns the interface of the M3 framework in Android mobile devices. The use of the M3 framework on Android devices has shown that it is not too consuming or unfeasible. In the second use case, we have shown how M3 framework can be integrated in a car dashboard. In the third use case, we have demonstrated an end-user centric approach which uses M3 embedded in smart fridges. In the fourth use case, we have demonstrated an end-user centric approach which uses M3 embedded in smart luggage. Finally, in the fifth use case, we have seen how the STAC security application, based on an interoperable cross-domain security knowledge base, can be used by developers to assist them in finding the security mechanisms fitting their needs to secure their IoT applications and architectures.

As a future work, we will integrate new use cases such as smart agriculture and smart energy. Adding more uses cases help us improve and make the M3 framework more generic.

In the next chapter (Chapter 7), we conclude this thesis and outline future work.

Chapter 7

Conclusion and Future Directions

"Choose a job you love, and you will never have to work a day in your life."

Confucius

"It always seems impossible until it's done."

Nelson Mandela

We conclude this thesis by summing up our contributions and by providing an outlook about the future research directions: (1) short-term challenges to improve our proposed Machine-to-Machine Measurements (M3) framework, and (2) long-term challenges to apply M3 in other domains such as quantum physics or neuroscience. Finally, we introduce social impacts of this thesis.

7.1 Conclusion

The motivation of this work has been driven by the need to make sensor data processing interoperable, to easily combine heterogeneous domains and build smarter IoT applications. In this thesis, we raised the following main challenge: **combining heterogeneous sensor data using semantic web technologies to design promising cross-domain IoT applications**. This huge challenge has been split into sub-challenges that we explained in section 2.4 and remind them hereafter (see Table 7.1):

- Challenge A: Interoperable IoT data has been addressed with the M3 nomenclature and ontology.
- Challenge B: Interpreting IoT data has been solved with S-LOR by reusing the domain knowledge referenced in LOV4IoT.
- Challenge C: Inter-domain interoperability with the M3 interoperable domain knowledge and LOV4IoT.

- Challenge D: Designing interoperable SWoT applications with the M3 framework and the SWoT generator.

- Challenge E: Securing IoT applications with STAC.

These challenges have been overcome through the contributions that we explain below.

Challenges	Proposed approaches
Challenge A: Interoperable IoT data	M3 nomenclature and ontology
Challenge B: Interpreting IoT data	S-LOR, LOV4IoT
Challenge C: Inter-domain interoperability + Reusing domain knowledge	M3 interoperable domain knowledge + LOV4IoT
Challenge D: Designing interoperable SWoT applications	SWoT generator, M3 framework
Challenge E: Securing IoT applications	STAC

Table 7.1: Challenges highlighted in the state of the art chapter overcame with the M3 framework

The first contribution is an innovative **Machine-to-Machine Measurements (M3) semantic-based framework** to assist developers in designing and implementing interoperable cross-domain IoT applications. The main novelty of M3 is to hide semantics to the developers. Further, using M3, machines can automatically understand high level information and with the intelligence embedded into them, they can act (control actuators, send notifications, etc.). M3 is composed of the SWoT generator which generates a M3 interoperable domain knowledge to easily design SWoT applications.

The second contribution is **Sensor-based Linked Open Rules (S-LOR)**, a novel approach to easily share, reuse and combine interoperable rules to infer high-level abstractions. S-LOR uses logical reasoning to provide semantic-based rules understandable by both humans and machines. Machines can automatically interpret IoT data to exchange and merge smarter data to build new IoT applications. This is a significant value compared to traditional approaches based on machine learning techniques; indeed, the rules can be shared and reused in other applications and are interoperable. Perera et al. explain in their survey that rule-based systems have less shortcomings than other approaches (supervised or unsupervised learning, fuzzy logic) [Perera et al., 2014]. Our proposed approach 'Sensor-based Linked Open Rules' can overcome the shortcomings explained in their survey: "rules should be defined manually, can be error prone and no validation or quality checking". Thanks to our innovative approach, such limitations could be easily overcome: (1) rules are designed in a interoperable manner to be shared and reused, and (2) they could be validated by domain experts. The M3 interoperable domain knowledge has been extracted from the Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset. LOV4IoT references, synthesizes and classifies more than 270 ontology-based projects relevant for IoT in various domains such as healthcare, transportation, smart home agriculture, smart energy, tourism, etc. The M3 framework has been integrated in an architecture compliant with ETSI M2M recommendations and is generic enough to be applicable in different

scenarios such as naturopathy, transportation & weather or tourism & weather. These scenarios have been inspired from the CityPulse[1] and IoT.est[2] scenarios and semantic-based projects referenced in LOV4IoT. Moreover, thanks to the flexibility and the maturity of the M3 framework, these scenarios can integrate M3 in different platforms: cloud, mobile devices and gateways.

The third contribution is **STAC (Security Toolbox: Attacks & Countermeasures**), a new cross-domain security application built using the same approach as M3 but in the security domain. The goal of STAC is to assist non-security expert developers in securing their software by suggesting the security mechanisms fitting their needs.

Table 7.2 reminds the major challenges highlighted in Section 2.4 and sums up the M3 components overcoming these challenges.

Research fields / Challenges	Internet of Things (IoT)	Machine-to-Machine (M2M)	Web of Things (WoT)	Semantic Web of Things (SWoT)	Semantic Sensor Networks (SSN)	M3 covers some limitations of these research fields
Interoperable IoT data	Ongoing	Ongoing	No	Yes	No. W3C SSN ontology is not enough	M3 nomenclature and ontology
Interpreting IoT data	Ongoing	Ongoing	No	Yes	Yes but not easy to reuse.	S-LOR, M3 rules
Inter-domain interoperability + Reusing domain knowledge	Ongoing	Ongoing	No	No	No	M3 interoperable domain knowledge + LOV4IoT
Designing interoperable SWoT applications	No	No	No	No	No	SWoT generator
Securing IoT	Ongoing	Ongoing	No	No	No	STAC

Table 7.2: Semantic Web of Things and related fields challenges are overcome with the M3 framework

To sum up, based on the semantics challenges introduced by Barnaghi et al. [Barnaghi et al., 2012b], semantics has been integrated in different levels in IoT thanks to the M3 framework which is composed of the following components: (1) data processing has been achieved with the M3 nomenclature and S-LOR, (2) reusing domain knowledge with LOV4IoT, (3) designing semantic-based services and applications with the SWoT generator, and (4) security with STAC.

[1] http://www.ict-citypulse.eu/scenarios/scenarios

[2] ict-iotest.eu/iotest/sites/default/files/files/public%20deliverables/IoT.est_D2.1_V1.0.pdf

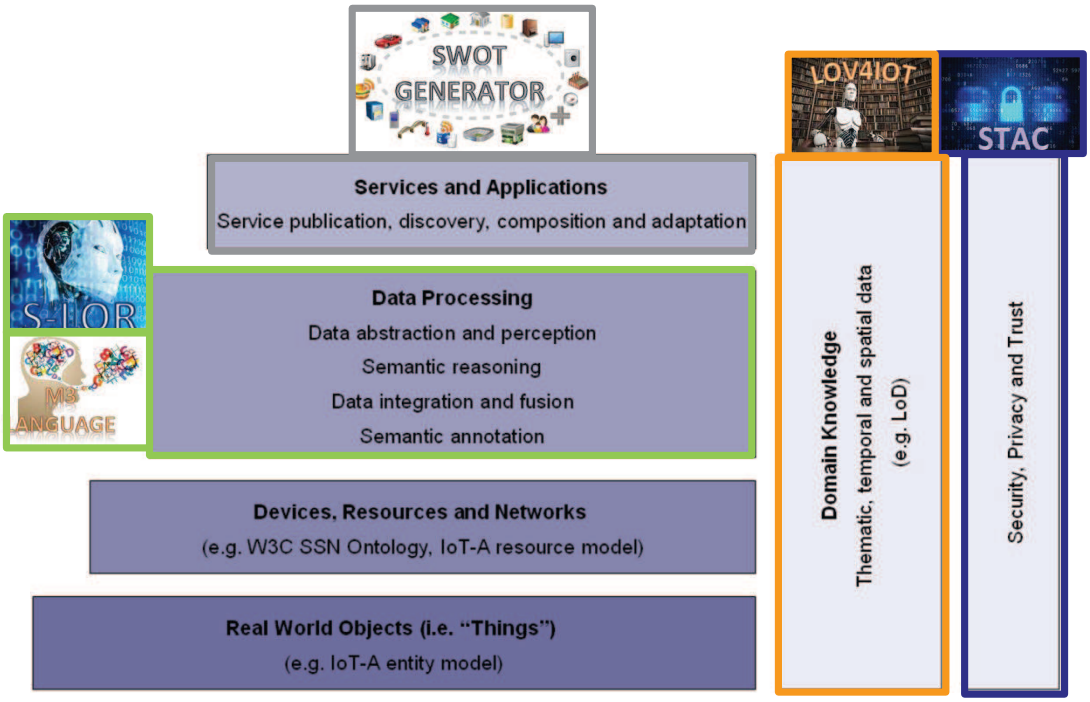

Figure 7.1: Semantic challenges in IoT overcome with M3

The M3 framework can be exploited by different stakeholders such as developers, domain experts, semantic web experts or commercial gateways, who can also enrich it. Most of the components of the M3 framework are available on our web site which has been visited more than 1771 times from 74 countries from December 5th 2013 to February 2015. Google Analytics has been integrated to our prototype since August 2014 and shows that LOV4IoT and the generation of Semantic Web of Things templates are the most visited pages on our website. Such results encourage to integrate semantics to Internet of Things.

7.2 Short Term Challenges, Future Directions and Discussions Regarding M3

In this section, we discuss the potential extensions of our work and the future challenges of Semantic Web of Things as depicted in Figure 7.2. In this picture, we synthesize and add new challenges to the ones introduced by Jara et al. [Jara et al., 2014] as follows:

- Merge the M3 framework to existing SWoT projects from the state of the art since they are complementary and not competitors.
- Standardize the M3 approach to describe sensor measurements to easily: (1) interpret

data, (2) combine domains, and (3) design interoperable SWoT applications.

- Sensor Plug & Play to automatically recognize devices (e.g., sensors or actuators) connected to M3 and get the produced data.
- Extract, reuse and combine the domain knowledge already available on the Web.

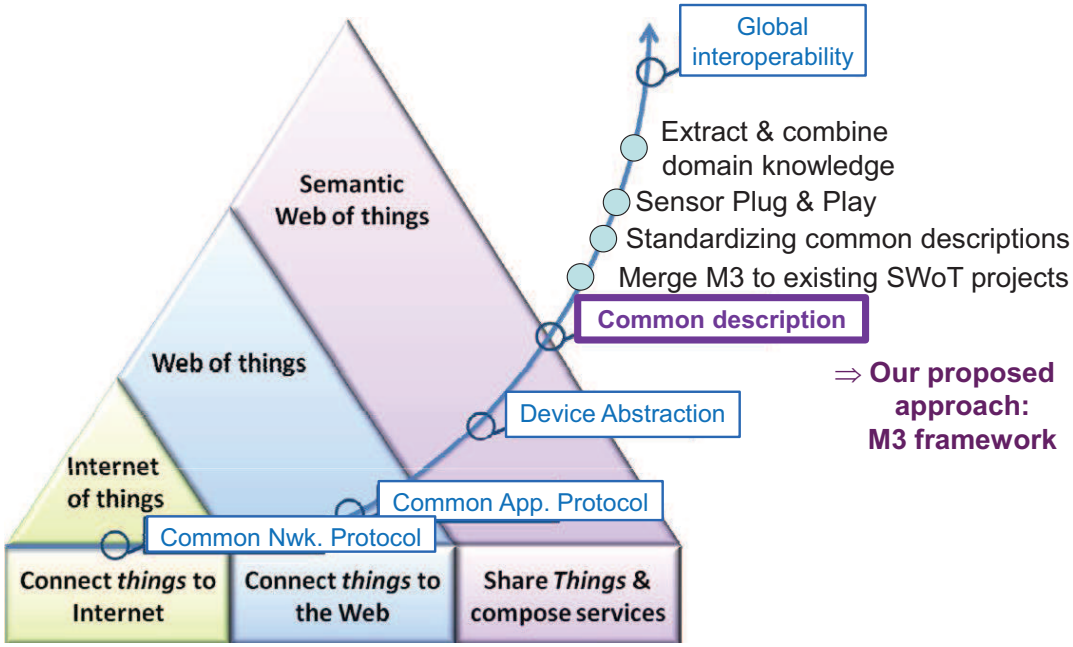

Figure 7.2: Semantic Web of Things future challenges

7.2.1 Synergizing efforts with standardization

A first step would be standardizing a common nomenclature to represent sensor measurements in an interoperable way. The proposed M3 nomenclature could be considered as a draft version for standardization. There is a real need to design common vocabularies and spread semantic web best practices in the Internet of Things community to easily share and reuse an interoperable domain knowledge. This will ease the process of interlinking domain knowledge to build promising cross-domain Semantic Web of Things applications.

Lessons learned during the tenure of the thesis have been disseminated to four standardization bodies or working groups: (1) semantic web guidelines to oneM2M [Gyrard and Bonnet, 2014], (2) the need for an interoperable domain knowledge to W3C Web of Things [Gyrard et al., 2014a], (3) the M3 ontology to ETSI M2M [Gyrard et al., 2013], and (4) S-LOR to W3C SSN ontology [Gyrard et al., 2014]. We aim to integrate M3 in an architecture that is compliant with oneM2M, as we have done it following the ETSI M2M standard. An essential step is to standardize domain ontologies in the same way W3C Time, W3C SSN ontology or

Schema.org already did. We suggested this idea to Schema.org[3]. Schema.org is a collection of vocabularies designed for popular search engines to enhance search results by structuring data on web sites. Such vocabularies are common and already embedded in popular tools generating web sites. Hence, search results can easily understand if the data referred to a person, a restaurant, opening hours, etc. Schema.org is not focused on IoT, thus, we need to provide a similar approach for IoT to structure and make sensor data interoperable to machines as it has been explained in this thesis.

7.2.2 Extracting the domain knowledge

The major challenge for LOV4IoT is to **automatically extract the domain knowledge** referenced in this dataset and redesign it in a interoperable manner. This will enable to **update automatically the knowledge bases** such as those of M3 or STAC. In order to achieve this challenge, there is a need to **improve semantic web tools to encourage best practices and better interoperability** to easily interlink the domain knowledge (ontologies, datasets or rules). The M3 knowledge base could be improved with new domains, sensors, measurements, units, rules and related domain knowledge. The STAC knowledge base could be updated with new technologies, security mechanisms and attacks.

The representation of the domain knowledge can be modeled in different manners. For instance, the driver's state can be modeled as a property or as a concept in the ontologies [Pollard et al., 2013] and [Fuchs et al., 2008b], and an apple can be defined as a concept[4] or an instance[5]. So, it is really difficult to say whether the way we modeled the M3 or the STAC domain knowledge is perfect or not; even if we have followed semantic web best practices as much as possible. Moreover, at the beginning of this work, we added numerous links between the M3 domain knowledge and existing domain ontologies. Due to the lack of the semantic web best practices, the Linked Open Vocabularies (LOV) community recommended us to avoid these links even if they are usually highly recommended since frequently, LOV bots found not dereferenceable URIs that hinder automation.

Currently, the LOV4IoT HTML user interface is static. It could become a dynamic web page based on the LOV4IoT RDF dataset. Users could choose only the domain they are interested in or the ontology status (e.g., only ontology shared online). Further, real-time analytic could be done easily to count the number of ontologies, etc.

7.2.3 Enhancing Sensor-based Linked Open Rules

The first major challenge for Linked Open Rules is to **automatically extract rules** from domain ontologies that we referenced in LOV4IoT to enrich S-LOR. OWL 2 RL[6] enables to detect owl:Restriction patterns, which perfectly fits our needs since most of the rules that we found are designed as owl:Restriction. We identified some implementations compatible with

[3] `http://schema.org/`

[4] `http://projects.kmi.open.ac.uk/smartproducts/ontologies/v2.6/food_taxonomy.owl`

[5] `http://www.sensormeasurement.appspot.com/dataset/naturopathy-dataset`

[6] `http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules`

our framework such as DLE Jena[7]. The second bigger challenge for the Linked Open Rules is to automatically combine domain knowledge using ontology matching tools. Due to the limitations of these tools, an open challenge is **improving, combining and designing new matching tools** tailored to our needs and to the domain ontologies referenced in LOV4IoT. Choosing and integrating ontology matching tools to automatically interlink cross-domain knowledge was time-consuming, exhausting and without success. We tried the automatic matching of ontologies with LogMap [Jiménez-Ruiz and Grau, 2011], Aroma [David, 2007], Anchor-Prompt [Noy and Musen, 2001], MAFRA [Maedche et al., 2002] tools and for matching datasets, the Silk tool [Volz et al., 2009b]. What is needed is to have some tools to assist beginners in finding the ontology matching tools fitting their needs and easily trying them. It is difficult to define the correct filter threshold value to select candidate mappings.

We also could propose the LOV4IoT dataset and the expected alignments as a new benchmark that highly differs from Ontology Alignment Evaluation Initiative (OAEI)[8]. OAEI is a benchmark frequently used to evaluate ontology matching tools. OAEI contains a set of ontologies to match and their structure differed from the ontologies referenced in the LOV4IoT dataset. For instance, ontologies referenced in OAEI contain a lot of properties associated to the classes which is not the case of the ontologies from LOV4IoT.

S-LOR is a logic-based approach to interpret sensor data and reason on them. We chose logical reasoning because it is easy to share and reuse rules to build the 'Linked Open Rules'. However, S-LOR has some limitations to interpret data generated by complex sensors such as accelerometer, electrocardiogram (ECG), etc. To deal with this limitation, S-LOR could be improved by combining semantic web technologies to machine learning to integrate more complicated rules such as the recognition of current activities of a user as proposed by Boshoven et al. [Boshoven and van Bommel, 2014]. We will search a way to share and reuse learning datasets. Another task would be to integrate into our framework either the KAT toolkit designed by Ganz et al. [Ganz et al., 2013] [Ganz et al., 2014] or the distributed approach of the Large Knowledge Collider (LarKC) project[9] [Fensel et al., 2008] which enables streaming reasoning.

7.2.4 Polishing the M3 framework

The M3 template dataset could be updated with new templates inspired by scenarios provided by CityPulse, IoT.est, oneM2M, ETSI M2M, etc. This will render the M3 framework more generic, since it will handle more IoT applications. The current implementation of the M3 converter to semantically annotate sensor measurements can be optimized by using other solution such as Extensible Stylesheet Language Transformations (XSLT). Such optimizations are essential if we have to handle the real-time scenarios too. Further, we could upgrade the converter to support all formats to ease the developer's task.

We could design M3 communications to enable smart discussions between things and applications. It could be based on "Semantic Web Services" approaches [McIlraith et al., 2001].

Inspired by the COMPOSE EU project [Mandler et al., 2013], we could provide the

[7] `http://lpis.csd.auth.gr/systems/DLEJena/`
[8] `http://oaei.ontologymatching.org/`
[9] `http://www.larkc.eu/`

'M3 open marketplace' to provide an app stores, a Software Development Kit (SDK) and an Integrated Development Environment (IDE) to support developers in designing Semantic Web of Things applications.

Cross-domain recommender system

Another challenge is to integrate a semantic and rule-based recommender system **to adapt the M3 suggestions to the user profile**. We found exciting works with common goals: cross-domain recommender systems based on semantic web technologies designed by [Hoxha, 2014] or [Tobías, 2013]. Another idea is to integrate and adapt to our needs the ontospread algorithm[10] [Rodríguez et al., 2013].

Sensor Plug & Play

Another challenge is to integrate the **discovery or Sensor Plug & Play** mechanism to our M3 framework to automatically recognize which sensors are plugged and to automatically build the application. To accomplish this last step, we could reuse and if necessary adapt the works proposed by [Datta et al., 2014a] [Datta et al., 2014b], [Bröring et al., 2011b], tools designed by the SPITFIRE project [Pfisterer et al., 2011], which uses the Global Sensor Networks (GSN) middleware [Aberer et al., 2006] [Calbimonte et al., 2014] (see Figure 7.3). We are strongly convinced that the M3 framework with some extensions such as Sensor Plug and Play and code refactoring could become a commercial product. If 'Sensor Plug & Play' is still too challenging, we could propose packages with popular hardware devices to automatically build and generate IoT applications.

Moreover, it would be very helpful, if the M3 framework could **consume sensor data** provided by other IoT projects. One solution could be to connect the M3 framework to the Graph of Things [Le-Phuoc et al., 2014] (which is already connected to GSN), to the CityPulse data (see Figure 7.3).

User interface

Regarding the user interfaces proposed by M3, it would be useful to automatically generate an user-friendly interface according to the type of sensors. This could be done by integrating works related to sensor data visualization [Logre et al., 2014] [Mosser et al., 2013]. These works are also based on the SenML format used to describe sensor measurements.

Enhancing security

An improvement of STAC that would be very helpful when designing secure IoT applications, is to **automatically integrate the security mechanism** suggested by STAC to a non secure IoT application. For instance, to secure data, we could automatically integrate

[10] `http://code.google.com/p/ontospread/`

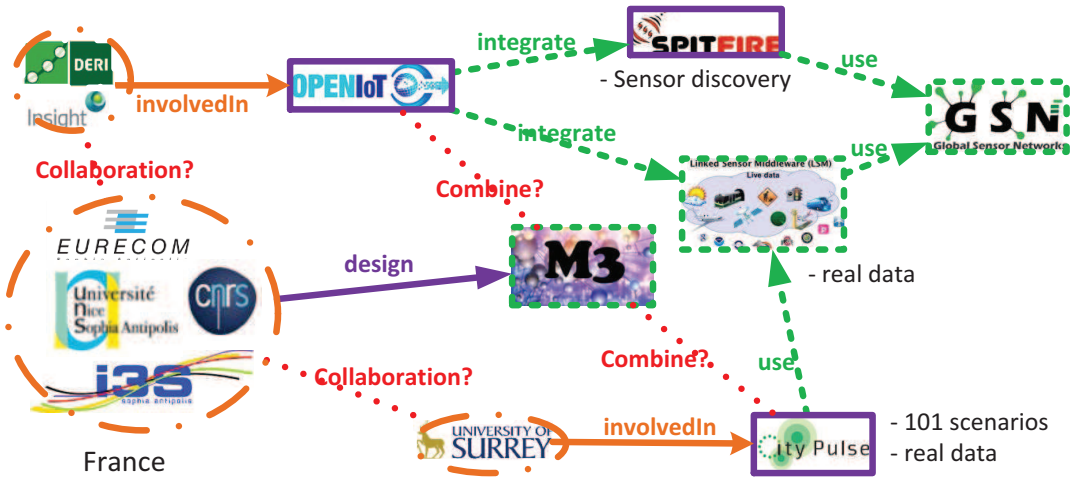

Figure 7.3: M3 connected to SWoT projects

the code employing cryptographic algorithms to encrypt and decrypt data. This improvement would be very useful when a **'security by design'** approach needs to be followed by non-security expert developers.

The mechanism of the 'Linked Open Rules' could be employed for the security domain too to check the current state of the system. For instance, to know if the system is under attack or potential attack. Indeed, Alam et al. underline the need of security reasoning for IoT through ontologies and semantic rules [Alam et al., 2011].

7.3 Long Term Challenges

Regarding 'Big Data', more broadly 'Data Management', the M3 approach could be proposed. Sharing only data is not enough, the added value is to interpret the data. A solution is not only share the data but also a package with both data, its models and the rules associated to these models as it has been done in the M3 approach. For instance, the abstraction of 'person' is constantly re-designed in all models. Theoretically, a new system just needs to reuse the models and rules to represent this abstraction and all related information, such as constraints on age (should be a number and superior to zero), the good format concerning the date, the string format is expected for the family name, etc. Such templates should be interoperable with each other, as is the case with the M3 interoperable domain knowledge and the M3 templates. The proposed M3 template could be adapted to the user's needs (e.g., size of data, frequency of measured data, raw or cleaned data). Thanks to the flexibility and modularity of our proposed approach, dealing with enormous quantities of sensor data should not be an issue. Actually, we could parallelize computations on data by classifying them by sensor measurement type or domain type, and then interlinking heterogeneous domains. The mechanism to generate M3 templates to interpret data could be

re-exploited in other domains such as quantum physics or neuroscience. Recently, data has been released by the Large Hadron Collider (LHC) at European Organization for Nuclear Research (CERN)[11]. However, without the related models and rules to interpret it, data is less valuable. Moreover, in neuroscience, interpreting brain waves data is challenging. Neuroscience experts explained "data may hold new insights into how the brain works but only if researchers can interpret it"[12]. In other complex domains such as neuroscience and quantum physics, it is essential to interpret data which is possible and easier with the associated models and rules. To resolve this issue, new M3 templates can be defined for new complex domains to share the way to model and interpret brain waves or quantum data.

7.4 Social impacts

The M3 framework will have social impacts in numerous domains. In healthcare, it enables improving the living conditions of elderly and people with deficiencies. They can still live in their own homes, the family or health unit can be alerted in case of hazard events. Meanwhile, it reduces cost hospitals, etc. In building automation, it is not just for improving comfort of inhabitants or employees, but to reduce energy consumption and save the planet. In smart transport, it enables to reduce traffic jams, the number of road accidents; it may save the lives of several people.

[11] http://home.web.cern.ch/about/updates/2014/11/cern-makes-public-first-data-lhc-experiments
[12] http://www.hhmi.org/news/new-tools-help-neuroscientists-analyze-big-data

Bibliography

[Abdelkader, 2009] Abdelkader, M. (2009). *Security and Service Provision Models for 4G Wireless Networks Submitted by: Manel Abdelkader*. PhD thesis, Engineering School of Communications, SupCom, Ariana, Tunisia.

[Aberer et al., 2006] Aberer, K., Hauswirth, M., and Salehi, A. (2006). 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.

[Abid et al., 2002] Abid, M., Sulistyo, S., and Najib, W. (2002). Umts security. security in core networks and utran.

[Abiteboul et al., 2012] Abiteboul, S., Antoine, E., and Stoyanovich, J. (2012). Viewing the web as a distributed knowledge base. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1–4. IEEE.

[Aggarwal et al., 2013] Aggarwal, C. C., Ashish, N., and Sheth, A. (2013). The internet of things: A survey from the data-centric perspective. In *Managing and mining sensor data*, pages 383–428. Springer.

[Ahmadian et al., 2009] Ahmadian, Z., Salimi, S., and Salahi, A. (2009). New attacks on umts network access. In *Wireless Telecommunications Symposium, 2009. WTS 2009*, pages 1–6. IEEE.

[Ahmed, 2009] Ahmed, A. S. (2009). An evaluation of security protocols on wireless sensor network.

[Aiash et al., 2010] Aiash, M., Mapp, G., Lasebae, A., and Phan, R. (2010). Providing security in 4g systems: unveiling the challenges. In *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*, pages 439–444. IEEE.

[Al-Massari, 2009] Al-Massari, M. M. (2009). Milenage: An example of good umts security.

[Al-Saraireh et al., 2006] Al-Saraireh, J., Yousef, S., and Al Nabhan, M. (2006). Enhancement mobile security and user confidentiality for umts. In *Second European Conference on Mobile Governement*.

[Alam et al., 2011] Alam, S., Chowdhury, M. M., and Noll, J. (2011). Interoperability of security-enabled internet of things. *Wireless Personal Communications*, 61(3):567–586.

[Alazeib and Diehl, 2005] Alazeib, A. and Diehl, A. (2005). An ontology for generic wireless authentication data. In *8th International Protege Conference-July*, pages 18–21.

[Ali and Kiefer, 2009] Ali, S. and Kiefer, S. (2009). μor–a micro owl dl reasoner for ambient intelligent devices. In *Advances in Grid and Pervasive Computing*, pages 305–316. Springer.

[Appliances, 2013] Appliances, T. S. (2013). Available semantics assets for the interoperability of smart appliances. mapping into a common ontology as a m2m application layer semantics.

[Arkko, 2012] Arkko, J. (2012). Network working group c. jennings internet-draft cisco intended status: Standards track z. shelby expires: January 18, 2013 sensinode. Media Type for Sensor Markup Language (SENML), draft-jennings-senml-09 (work in progress).

[Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

[Avancha et al., 2004] Avancha, S., Patel, C., and Joshi, A. (2004). Ontology-driven adaptive sensor networks. In *First Annual International Conference on Mobile and Ubiquitous Systems, Networking and Services*, pages 194–202.

[Bandyopadhyay and Sen, 2011] Bandyopadhyay, D. and Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69.

[Barbero et al., 2011] Barbero, C., Zovo, P. D., and Gobbi, B. (2011). A flexible context aware reasoning approach for iot applications. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, volume 1, pages 266–275. IEEE.

[Barbieri et al., 2009] Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., and Grossniklaus, M. (2009). C-sparql: Sparql for continuous querying. In *Proceedings of the 18th international conference on World wide web*, pages 1061–1062. ACM.

[Barnaghi et al., 2013] Barnaghi, P., Cousin, P., Malo, P., Serrano, M., and Viho, C. (2013). Simpler iot word (s) of tomorrow, more interoperability challenges to cope today. *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*, page 277.

[Barnaghi et al., 2011] Barnaghi, P., Ganz, F., Abangar, H., Presser, M., and Moessner, K. (2011). Sense2web: A linked data platform for semantic sensor networks.

[Barnaghi et al., 2012a] Barnaghi, P., Ganz, F., Henson, C., and Sheth, A. (2012a). Computing perception from sensor data.

[Barnaghi et al., 2009] Barnaghi, P., Meissner, S., Presser, M., and Moessner, K. (2009). Sense and sens'ability: Semantic data modelling for sensor networks. In *Conference Proceedings of ICT Mobile Summit 2009*. purl.oclc.org/net/unis/ontology/sensordata.owl.

[Barnaghi et al., 2010] Barnaghi, P., Presser, M., and Moessner, K. (2010). Publishing linked sensor data. In *CEUR Workshop Proceedings: Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN), Organised in conjunction with the International Semantic Web Conference*, volume 668.

[Barnaghi et al., 2014] Barnaghi, P., Tönjes, R., Höller, J., Hauswirth, M., Sheth, A., and Anantharam, P. (2014). Citypulse: Real-time iot stream processing and large-scale data analytics for smart city applications. In *Europen Semantic Web Conference (ESWC) 2014*.

[Barnaghi et al., 2012b] Barnaghi, P., Wang, W., Henson, C., and Taylor, K. (2012b). Semantics for the internet of things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(1):1–21.

[Baumgartner and Retschitzegger, 2006] Baumgartner, N. and Retschitzegger, W. (2006). A survey of upper ontologies for situation awareness. In *Proc. of the 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering, St. Thomas, US VI*, pages 1–9.

[Becker and Bizer, 2008] Becker, C. and Bizer, C. (2008). Dbpedia mobile: A location-enabled linked data browser. *LDOW*, 369.

[Beji and El Kadhi, 2009] Beji, S. and El Kadhi, N. (2009). Security ontology proposal for mobile applications. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 580–587. IEEE.

[Bell et al., 2009] Bell, D., Heravi, B., and Lycett, M. (2009). Sensory semantic user interfaces (sensui). In *2nd International Workshop on Semantic Sensor Networks 2009, Washington, Oct 2009*.

[Bermejo et al., 2013] Bermejo, A., Villadangos, J., Astrain, J., and Cordoba, A. (2013). Ontology based road traffic management. In *Intelligent Distributed Computing VI*, pages 103–108. Springer.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

[Berrueta et al., 2008] Berrueta, D., Fernández, S., and Frade, I. (2008). Cooking http content negotiation with vapour. In *Proceedings of 4th Workshop on Scripting for the Semantic Web (SFSW2008)*. Citeseer.

[Bettini et al., 2010] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180.

[Bikakis et al., 2008] Bikakis, A., Patkos, T., Antoniou, G., and Plexousakis, D. (2008). A survey of semantics-based approaches for context reasoning in ambient intelligence. In *Constructing ambient intelligence*, pages 14–23. Springer.

[Bizer et al., 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22. http://www.w3.org/DesignIssues/LinkedData.html.

[Blanco-Fernández et al., 2011] Blanco-Fernández, Y., López-Nores, M., Pazos-Arias, J. J., García-Duque, J., and Martín-Vicente, M. I. (2011). Tripfromtv+: Targeting personalized tourism to interactive digital tv viewers by social networking and semantic reasoning. *Consumer Electronics, IEEE Transactions on*, 57(2):953–961.

[Blansché et al., 2010] Blansché, A., Cojan, J., Dufour-Lussier, V., Lieber, J., Molli, P., Nauer, E., Skaf-Molli, H., and Toussaint, Y. (2010). Taaable 3: Adaptation of ingredient quantities and of textual preparations. In *18th Int. Conf. on Case-Based Reasoning Workshop Procs*, pages 189–198. Citeseer.

[Boman et al., 2002] Boman, K., Horn, G., Howard, P., and Niemi, V. (2002). Umts security. *Electronics & Communication Engineering Journal*, 14(5):191–204.

[Bonino and Corno, 2008] Bonino, D. and Corno, F. (2008). Dogont-ontology modeling for intelligent domotic environments. *The Semantic Web-ISWC 2008*, pages 790–803.

[Bonino and Corno, 2010] Bonino, D. and Corno, F. (2010). Rule-based intelligence for domotic environments. *Automation in Construction*, 19(2):183–196.

[Bonino et al., 2014a] Bonino, D., Corno, F., and De Russis, L. (2014a). Poweront: An ontology-based approach for power consumption estimation in smart homes. In *1st Cognitive Internet of Things Technologies (COIOTE 2014)*.

[Bonino et al., 2014b] Bonino, D., Corno, F., and De Russis, L. (2014b). A semantics-rich information technology architecture for smart buildings. *Buildings*, 4(4):880–910.

[Bontas et al., 2005] Bontas, E. P., Mochol, M., and Tolksdorf, R. (2005). Case studies on ontology reuse. In *Proceedings of the IKNOW05 International Conference on Knowledge Management*, volume 74.

[Booysen et al., 2012] Booysen, M., Gilmore, J., Zeadally, S., and Van Rooyen, G. (2012). Machine-to-machine (m2m) communications in vehicular networks. In *Article*. Korea Society of Internet Information (KSII).

[Borgohain et al., 2015] Borgohain, T., Kumar, U., and Sanyal, S. (2015). Survey of security and privacy issues of internet of things. *arXiv preprint arXiv:1501.02211*.

[Bormann et al., 2012] Bormann, C., Castellani, A. P., and Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2):62–67.

[Borst, 1997] Borst, W. N. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente.

[Boshoven and van Bommel, 2014] Boshoven, B. and van Bommel, P. (2014). Personalized life reasoning. Master's thesis.

[Boswarthick et al., 2012] Boswarthick, D., Elloumi, O., and Hersent, O. (2012). *M2M communications: a systems approach*. Wiley.

[Botts et al., 2008] Botts, M., Percivall, G., Reed, C., and Davidson, J. (2008). Ogc® sensor web enablement: Overview and high level architecture. *GeoSensor networks*, pages 175–190. http://www.opengeospatial.org/projects/groups/sensorweb.

[Boyle and Newe, 2007] Boyle, P. and Newe, T. (2007). Security protocols for use with wireless sensor networks: A survey of security architectures. In *Wireless and Mobile Communications, 2007. ICWMC'07. Third International Conference on*, pages 54–54. IEEE.

[Bradeško et al., 2012] Bradeško, L., Moraru, A., Fortuna, B., Fortuna, C., and Mladenić, D. (2012). A framework for acquiring semantic sensor descriptions (short paper). *Semantic Sensor Networks*, page 97.

[Bröring et al., 2011a] Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., and Lemmens, R. (2011a). New generation sensor web enablement. *Sensors*, 11(3):2652–2699.

[Bröring et al., 2011b] Bröring, A., Maué, P., Janowicz, K., Nüst, D., and Malewski, C. (2011b). Semantically-enabled sensor plug & play for the sensor web. *Sensors*, 11(8):7568–7605.

[Cabral et al., 2014] Cabral, L., Compton, M., and Müller, H. (2014). A use case in semantic modelling and ranking for the sensor web. In *The Semantic Web–ISWC 2014*, pages 276–291. Springer.

[Calbimonte, 2013] Calbimonte, J.-P. (2013). *Ontology-based access to sensor data streams*. PhD thesis, Informatica.

[Calbimonte et al., 2010] Calbimonte, J.-P., Corcho, O., and Gray, A. J. (2010). Enabling ontology-based access to streaming data sources. In *The Semantic Web–ISWC 2010*, pages 96–111. Springer.

[Calbimonte et al., 2014] Calbimonte, J.-P., Sarni, S., Eberle, J., and Aberer, K. (2014). Xgsn: An open-source semantic sensing middleware for the web of things. In *7th International Workshop on Semantic Sensor Networks*, number EPFL-CONF-200926.

[Calder et al., 2010] Calder, M., Morris, R., and Peri, F. (2010). Machine reasoning about anomalous sensor data. *Ecological Informatics*, 5(1):9–18. http://www.cesn.org/sensor/cesn.owl.

[Cameron et al., 2009] Cameron, M. A., Wu, J., Taylor, K., Ratcliffe, D., Squire, G., Colton, J., Taylor, K., Ayyagari, A., and De Roure, D. (2009). Semantic solutions for integration of federated ocean observations. In *SSN*, pages 64–79. Citeseer.

[Cantais et al., 2005] Cantais, J., Dominguez, D., Gigante, V., Laera, L., and Tamma, V. (2005). 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*.

[Caragata et al., 2011a] Caragata, D., Assad, S., Tutanescu, I., Shoniregun, C. A., and Akmayeva, G. (2011a). Security of mobile internet access with umts/hsdpa/lte. In *Internet Security (WorldCIS), 2011 World Congress on*, pages 272–276. IEEE.

[Caragata et al., 2011b] Caragata, D., El Assad, S., Shoniregun, C., and Akmayeva, G. (2011b). Umts security: Enhancement of identification, authentication and key agreement protocols. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 278–282. IEEE.

[Casado and Tsigas, 2009] Casado, L. and Tsigas, P. (2009). Contikisec: A secure network layer for wireless sensor networks under the contiki operating system. In *Identity and Privacy in the Internet Age*, pages 133–147. Springer.

[Cassou et al., 2012] Cassou, D., Bruneau, J., Consel, C., and Balland, E. (2012). Toward a tool-based development methodology for pervasive computing applications. *Software Engineering, IEEE Transactions on*, 38(6):1445–1463.

[Chaari et al., 2007] Chaari, T., Ejigu, D., Laforest, F., and Scuturici, V.-M. (2007). A comprehensive approach to model and use context for adapting applications in pervasive environments. *Journal of Systems and Software*, 80(12):1973–1992.

[Chahuara et al., 2012] Chahuara, P., Portet, F., and Vacher, M. (2012). Context aware decision system in a smart home: knowledge representation and decision making using uncertain contextual information. In *The 4th International Workshop on Acquisition, Representation and Reasoning with Contextualized Knowledge (ARCOE-12)*, pages 52–64.

[Chen, 2003] Chen, H. (2003). An intelligent broker architecture for context-aware systems. *PhD proposal in Computer Science, University of Maryland, Baltimore, USA*.

[Chen et al., 2003] Chen, H., Finin, T., and Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03):197–207.

[Chen et al., 2005a] Chen, H., Finin, T., and Joshi, A. (2005a). Semantic web in the context broker architecture. Technical report, DTIC Document.

[Chen et al., 2005b] Chen, H., Finin, T., and Joshi, A. (2005b). The soupa ontology for pervasive computing. In *Ontologies for agents: Theory and experiences*, pages 233–258. Springer.

[Chen et al., 2004] Chen, H., Perich, F., Finin, T., and Joshi, A. (2004). Soupa: Standard ontology for ubiquitous and pervasive applications. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 258–267. IEEE.

[Chen, 2004] Chen, H. L. (2004). *An intelligent broker architecture for pervasive context-aware systems*. PhD thesis, University of Maryland, Baltimore County.

[Chen and Nugent, 2009] Chen, L. and Nugent, C. (2009). Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430.

[Chen et al., 2013] Chen, L., Nugent, C., and Rafferty, J. (2013). Ontology-based activity recognition framework and services. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, page 463. ACM.

[Chen et al., 2014] Chen, S., Xu, H., Liu, D., Hu, B., and Wang, H. (2014). A vision of iot applications challenges and opportunities with china perspective.

[Chen et al., 2006] Chen, Y.-H., Huang, T.-h., Hsu, D. C., and Hsu, J. Y.-j. (2006). Colorcocktail: an ontology-based recommender system. *Proceedings of 20th American Association for Artificial Intelligence. Menlo Park, USA: AAAI Press*, pages 79–82.

[Cheng et al., 2012] Cheng, Y., Naslund, M., Selander, G., and Fogelstrom, E. (2012). Privacy in machine-to-machine communications a state-of-the-art survey. In *Communication Systems (ICCS), 2012 IEEE International Conference on*, pages 75–79. IEEE.

[Chien et al., 2013] Chien, H.-Y., Chen, S.-K., Lin, C.-Y., Yan, J.-L., Liao, W.-C., Chu, H.-Y., Chen, K.-J., Lai, B.-F., and Chen, Y.-T. (2013). Design and implementation of zigbee-ontology-based exhibit guidance and recommendation system. *International Journal of Distributed Sensor Networks*, 2013.

[Choi et al., 2009] Choi, C., Cho, M., Choi, J., Hwang, M., Park, J., and Kim, P. (2009). Travel ontology for intelligent recommendation system. In *Modelling & Simulation, 2009. AMS'09. Third Asia International Conference on*, pages 637–642. IEEE.

[Christopoulou et al., 2004] Christopoulou, E., Goumopoulos, C., Zaharakis, I., and Kameas, A. (2004). An ontology-based conceptual model for composing context-aware applications. *Research Academic Computer Technology Institute*.

[Christopoulou and Kameas, 2005] Christopoulou, E. and Kameas, A. (2005). Gas ontology: an ontology for collaboration among ubiquitous computing devices. *International Journal of Human-Computer Studies*, 62(5):664–685.

[Codina and Ceccaroni, 2010a] Codina, V. and Ceccaroni, L. (2010a). A recommendation system for the semantic web. In *Distributed Computing and Artificial Intelligence*, pages 45–52. Springer.

[Codina and Ceccaroni, 2010b] Codina, V. and Ceccaroni, L. (2010b). Taking advantage of semantics in recommendation systems. In *Proceedings of the 2010 Conference on Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*, pages 163–172, Amsterdam, The Netherlands, The Netherlands. IOS Press.

[Codina and Ceccaroni, 2010c] Codina, V. and Ceccaroni, L. (2010c). Taking advantage of semantics in recommendation systems. In *Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*, volume 220, page 163. IOS Press.

[Codina et al., 2013] Codina, V., Ricci, F., and Ceccaroni, L. (2013). Semantically-enhanced pre-filtering for context-aware recommender systems. In *Proceedings of the 3rd Workshop on Context-awareness in Retrieval and Recommendation*, pages 15–18. ACM.

[Compton et al., 2012] Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*. http://www.w3.org/2005/Incubator/ssn/ssnx/ssn.

[Compton et al., 2009] Compton, M., Henson, C., Lefort, L., Neuhaus, H., and Sheth, A. (2009). A survey of the semantic specification of sensors. *Proc. Semantic Sensor Networks*, 17.

[Coppens et al., 2013] Coppens, S., Vander Sande, M., Verborgh, R., Mannens, E., and Van de Walle, R. (2013). Reasoning over sparql. In *Proceedings of the 6th Workshop on Linked Data on the Web*.

[Corcho and García-Castro, 2010] Corcho, O. and García-Castro, R. (2010). Five challenges for the semantic sensor web. *Semantic Web*, 1(1):121–125.

[Costabello, 2013] Costabello, L. (2013). *Context-aware access control and presentation of linked data*. PhD thesis, Université Nice Sophia Antipolis.

[Coyle et al., 2007] Coyle, L., Neely, S., Stevenson, G., Sullivan, M., Dobson, S., Nixon, P., and Rey, G. (2007). Sensor fusion-based middleware for smart homes. *International Journal of Assistive Robotics and Mechatronics*, 8(2):53–60.

[Damljanović and Devedzic, 2008] Damljanović, D. and Devedzic, V. (2008). Applying semantic web to e-tourism. *The Semantic Web for Knowledge and Data Management: Technologies and Practices. Information Science Reference (IGI Global)*.

[Damljanovic and Devedzic, 2008] Damljanovic, D. and Devedzic, V. (2008). Semantic web and e-tourism. *Encyclopedia of Information Science and Technology, Second edition. IGI Global*.

[d'Aquin and Motta, 2011] d'Aquin, M. and Motta, E. (2011). Watson, more than a semantic web search engine. *Semantic Web*, 2(1):55–63. http://watson.kmi.open.ac.uk/WatsonWUI/.

[d'Aquin et al., 2011] d'Aquin, M., Nikolov, A., and Motta, E. (2011). Building sparql-enabled applications with android devices. In *10th International Semantic Web Conference (ISWC2011), 23-27 Oct 2011, Bonn, Germany*.

[Daramola et al., 2009] Daramola, O., Adigun, M., and Ayo, C. (2009). Building an ontology-based framework for tourism recommendation services. *Information and Communication Technologies in Tourism 2009*, pages 135–147.

[Datta et al., 2014a] Datta, S., Bonnet, C., and Nikaein, N. (2014a). Cct: Connect and control things: A novel mobile application to manage m2m devices and endpoints. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6.

[Datta, 2015] Datta, S. K. (2015). M2M communications and Internet of Things as enablers of smart city.

[Datta et al., 2014b] Datta, S. K., Bonnet, C., and Nikaein, N. (2014b). An iot gateway centric architecture to provide novel m2m services. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 514–519. IEEE.

[David, 2007] David, J. (2007). Association rule ontology matching approach. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(2):27–49.

[David et al., 2012a] David, J., Euzenat, J., Rosoiu, M., et al. (2012a). Linked data from your pocket. In *Proc. 1st ESWC workshop on downscaling the semantic web*, pages 6–13.

[David et al., 2012b] David, J., Euzenat, J., Rosoiu, M., et al. (2012b). Mobile api for linked data.

[David Wood, 2014] David Wood, Marsha Zaidman, L. R. e. M. H. (2014). *Linked Data. Structured Data on the Web.*

[De et al., 2012] De, S., Elsaleh, T., Barnaghi, P., and Meissner, S. (2012). An internet of things platform for real-world and digital objects. *Scalable Computing: Practice and Experience*, 13(1).

[Dema, 2008] Dema, T. (2008). *eTourPlan: A knowledge-based tourist route and activity planner*. PhD thesis, UNIVERSITY OF NEW BRUNSWICK.

[Denissen et al., 2008] Denissen, J. J., Butalid, L., Penke, L., and Van Aken, M. A. (2008). The effects of weather on daily mood: a multilevel approach. *Emotion*, 8(5):662.

[Denker et al., 2003] Denker, G., Kagal, L., Finin, T., Paolucci, M., and Sycara, K. (2003). Security for daml web services: Annotation and matchmaking. *The Semantic Web-ISWC 2003*, pages 335–350.

[Denker et al., 2004] Denker, G., Nguyen, S., and Ton, A. (2004). Owl-s semantics of security web services: A case study. *The Semantic Web: Research and Applications*, pages 240–253.

[Desai et al., 2014] Desai, P., Sheth, A., and Anantharam, P. (2014). Semantic gateway as a service architecture for iot interoperability. *arXiv preprint arXiv:1410.4977*.

[Deshmukh et al., 2011] Deshmukh, S. V., Radake, D. P., and Hande, K. N. (2011). Driver fatigue detection using sensor network. *Int. J. Eng. Sci. Technol*, pages 89–92.

[Devaraju and Kauppinen, 2012] Devaraju, A. and Kauppinen, T. (2012). Sensors tell more than they sense: Modeling and reasoning about sensor observations for understanding weather events. *International Journal of Sensors Wireless Communications and Control*, 2(1):14–26.

[Douceur, 2002] Douceur, J. R. (2002). The sybil attack. In *Peer-to-peer Systems*, pages 251–260. Springer.

[Eid et al., 2006] Eid, M., Liscano, R., and El Saddik, A. (2006). A novel ontology for sensor networks data. In *Computational Intelligence for Measurement Systems and Applications, Proceedings of 2006 IEEE International Conference on*, pages 75–79. IEEE.

[Eid et al., 2007] Eid, M., Liscano, R., and El Saddik, A. (2007). 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.

[Ejigu et al., 2007] Ejigu, D., Scuturici, M., and Brunie, L. (2007). An ontology-based approach to context modeling and reasoning in pervasive computing. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pages 14–19. IEEE.

[Ekelhart et al., 2009] Ekelhart, A., Fenz, S., and Neubauer, T. (2009). Aurum: A framework for information security risk management. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–10. IEEE.

[Erschbamer and Malleier, 2012] Erschbamer, J. and Malleier, M. (2012). A knowledge-based menu recommender system. *Free University of Bolzano*.

[Euzenat, 2011] Euzenat, J. (2011). Semantic technologies and ontology matching for interoperability inside and across buildings. In *Proc. 2nd CIB workshop on eeBuildings data models*, pages 22–34.

[Euzenat et al., 2008] Euzenat, J., Pierson, J., and Ramparany, F. (2008). Dynamic context management for pervasive applications. *The Knowledge Engineering Review*, 23(01):21–49.

[Euzenat and Shvaiko, 2013] Euzenat, J. and Shvaiko, P. (2013). *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition.

[Euzenat, 2014] Euzenat, J. e. a. (2014). Deliverable d2.2: Ontologies and datasets for energy management system interoperability v1.

[Evesti et al., 2011] Evesti, A., Savola, R., Ovaska, E., and Kuusijärvi, J. (2011). The design, instantiation, and usage of information security measuring ontology. In *MOPAS 2011, The Second International Conference on Models and Ontology-based Design of Protocols, Architectures and Services*, pages 1–9.

[Fensel et al., 2008] Fensel, D., van Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Della Valle, E., Fischer, F., Huang, Z., Kiryakov, A., Lee, T.-I., et al. (2008). Towards larkc: a platform for web-scale reasoning. In *Semantic Computing, 2008 IEEE International Conference on*, pages 524–529. IEEE.

[Fernández-López et al., 1997] Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

[Figueiras et al., 2013] Figueiras, P., Costa, R., Malo, P., Bradesko, L., and Jermol, M. (2013). Knowledge base approach for developing a mobile personalized travel companion. In *ITS Telecommunications (ITST), 2013 13th International Conference on*, pages 97–103. IEEE.

[Freyne et al., 2011] Freyne, J., Berkovsky, S., and Smith, G. (2011). Recipe recommendation: accuracy and reasoning. In *User Modeling, Adaption and Personalization*, pages 99–110. Springer.

[Frye et al., 2012] Frye, L., Cheng, L., and Heflin, J. (2012). An ontology-based system to identify complex network attacks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6683–6688. IEEE.

[Fuchs et al., 2008a] Fuchs, S., Rass, S., and Kyamakya, K. (2008a). Integration of ontological scene representation and logic-based reasoning for context-aware driver assistance systems. *Electronic Communications of the EASST*, 11.

[Fuchs et al., 2008b] Fuchs, S., Rass, S., Lamprecht, B., and Kyamakya, K. (2008b). A model for ontology-based scene description for context-aware driver assistance systems. In *Proceedings of the 1st international conference on Ambient media and systems*, page 5. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Fudholi et al., 2009] Fudholi, D. H., Maneerat, N., Varakulsiripunth, R., and Kato, Y. (2009). Application of protégé, swrl and sqwrl in fuzzy ontology-based menu recommendation. In *Intelligent Signal Processing and Communication Systems, 2009. ISPACS 2009. International Symposium on*, pages 631–634. IEEE.

[Gaire et al., 2013] Gaire, R., Lefort, L., Compton, M., Falzon, G., Lamb, D., and Taylor, K. (2013). Demonstration: Semantic web enabled smart farm with gsn. In *International Semantic Web Conference (Posters & Demos)*, pages 41–44.

[Ganz, 2014] Ganz, F. (2014). *Intelligent Communication and Information Processing for Cyber-Physical Data*. PhD thesis, University of Surrey.

[Ganz et al., 2013] Ganz, F., Barnaghi, P., and Carrez, F. (2013). Information abstraction for heterogeneous real world internet data. *Sensors Journal, IEEE*, 13(10):3793–3805.

[Ganz et al., 2014] Ganz, F., Barnaghi, P., and Carrez, F. (2014). Automated semantic knowledge acquisition from sensor data. *Systems Journal, IEEE*, PP(99):1–12.

[Ganz et al., 2015] Ganz, F., Puschmann, D., Barnaghi, P., and Carrez, F. (2015). A practical evaluation of information processing and abstraction techniques for the internet of things. *IEEE Internet of Things journal*.

[García-Crespo et al., 2011] García-Crespo, Á., López-Cuadrado, J. L., Colomo-Palacios, R., González-Carrasco, I., and Ruiz-Mezcua, B. (2011). Sem-fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert systems with applications*, 38(10):13310–13319.

[Gluhak et al., 2011] Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., and Razafindralambo, T. (2011). A survey on facilities for experimental internet of things research. *Communications Magazine, IEEE*, 49(11):58–67.

[Gómez-Goiri et al., 2014] Gómez-Goiri, A., Orduña, P., Diego, J., and López-de Ipiña, D. (2014). Otsopack: Lightweight semantic framework for interoperable ambient intelligence applications. *Computers in Human Behavior*, 30:460–467.

[Goodwin and Russomanno, 2006] Goodwin, C. and Russomanno, D. J. (2006). An ontology-based sensor network prototype environment. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN 2006), Nashville TN, USA*.

[Gray et al., 2011] Gray, A. J., García-Castro, R., Kyzirakos, K., Karpathiotakis, M., Calbimonte, J.-P., Page, K., Sadler, J., Frazer, A., Galpin, I., Fernandes, A. A., et al. (2011). A semantically enabled service architecture for mashups over streaming and stored data. In *The Semanic Web: Research and Applications*, pages 300–314. Springer.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

[Gu and Wang, 2009] Gu, H. and Wang, D. (2009). A content-aware fridge based on rfid in smart home for home-healthcare. In *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, volume 2, pages 987–990. IEEE.

[Gu et al., 2004] Gu, T., Wang, X. H., Pung, H. K., and Zhang, D. Q. (2004). An ontology-based context model in intelligent environments. In *Proceedings of communication networks and distributed systems modeling and simulation conference*, volume 2004, pages 270–275.

[Gubbi et al., 2013] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

[Guinard et al., 2010] Guinard, D., Trifa, V., and Wilde, E. (2010). A resource oriented architecture for the web of things. In *Internet of Things (IOT), 2010*, pages 1–8. IEEE.

[Gupta and Patil, 2009] Gupta, P. and Patil, P. (2009). 4g-a new era in wireless telecommunication. *Magister Program in S/W Engineering, Malardalen University*.

[Gyrard and Bonnet, 2014] Gyrard, A. and Bonnet, C. (2014). Semantic Web best practices: Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things.

[Gyrard et al., 2013] Gyrard, A., Bonnet, C., and Boudaoud, K. (2013). An ontology to semantically annotate the M2M data.

[Gyrard et al., 2013] Gyrard, A., Bonnet, C., and Boudaoud, K. (2013). The stac (security toolbox: attacks & countermeasures) ontology. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 165–166. International World Wide Web Conferences Steering Committee.

[Gyrard et al., 2014a] Gyrard, A., Bonnet, C., and Boudaoud, K. (2014a). Domain knowledge Interoperability to build the semantic web of things.

[Gyrard et al., 2014b] Gyrard, A., Bonnet, C., and Boudaoud, K. (2014b). Enrich machine-to-machine data with semantic web technologies for cross-domain applications. In *WF-IOT 2014, World Forum on Internet of Things, 6-8 March 2014, Seoul, Korea*, Seoul, KOREA, REPUBLIC OF.

[Gyrard et al., 2014] Gyrard, A., Bonnet, C., and Boudaoud, K. (2014). Helping IoT application developers with sensor-based linked open rules. In *SSN 2014, 7th International Workshop on Semantic Sensor Networks in conjunction with the 13th International Semantic Web Conference (ISWC 2014), 19-23 October 2014, Riva Del Garda, Italy.*

[Haase and Wang, 2007] Haase, P. and Wang, Y. (2007). A decentralized infrastructure for query answering over distributed ontologies. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1351–1356. ACM.

[Hachem, 2014] Hachem, S. (2014). *Service-Oriented Middleware for the Large-Scale Mobile Internet of Thingd*. PhD thesis, Université de Versailles-Saint Quentin en Yvelines.

[Hachem et al., 2011] Hachem, S., Teixeira, T., and Issarny, V. (2011). Ontologies for the internet of things. In *Proceedings of the 8th Middleware Doctoral Symposium*, page 3. ACM.

[Hadzic et al., 2008] Hadzic, M., Chen, M., and Dillon, T. S. (2008). Towards the mental health ontology. In *Bioinformatics and Biomedicine, 2008. BIBM'08. IEEE International Conference on*, pages 284–288. IEEE.

[Hamid et al., 2006] Hamid, M. A., Mamun-Or-Rashid, M., and Hong, C. S. (2006). Routing security in sensor network: Hello flood attack and defense. *IEEE ICNEWS*, pages 2–4.

[Hastings et al., 2011] Hastings, J., Ceusters, W., Smith, B., and Mulligan, K. (2011). The emotion ontology: enabling interdisciplinary research in the affective sciences. *Modeling and Using Context*, pages 119–123.

[Heath and Bizer, 2011] Heath, T. and Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

[Henricksen and Indulska, 2006] Henricksen, K. and Indulska, J. (2006). Developing context-aware pervasive computing applications: Models and approach. *Pervasive and mobile computing*, 2(1):37–64.

[Henson et al., 2009] Henson, C., Pschorr, J., Sheth, A., and Thirunarayan, K. (2009). Semsos: Semantic sensor observation service. In *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, pages 44–53. IEEE.

[Henson et al., 2012] Henson, C., Sheth, A., and Thirunarayan, K. (2012). Semantic perception: Converting sensory observations to abstractions. *Internet Computing, IEEE*, 16(2):26–34.

[Henson, 2013] Henson, C. A. (2013). *A Semantics-based Approach to Machine Perception*. PhD thesis, Wright State University.

[Hervás et al., 2013] Hervás, R., Fontecha, J., Ausín, D., Castanedo, F., Bravo, J., and López-de Ipiña, D. (2013). Mobile monitoring and reasoning methods to prevent cardiovascular diseases. *Sensors*, 13(5):6524–6541.

[Herzog et al., 2007] Herzog, A., Shahmehri, N., and Duma, C. (2007). An ontology of information security. *International Journal of Information Security and Privacy (IJISP)*, 1(4):1–23.

[Hilera and Ruiz, 2006] Hilera, J. R. and Ruiz, F. (2006). Ontologies in ubiquitous computing. In *ICUC*.

[Horrocks et al., 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79.

[Hoxha, 2014] Hoxha, J. (2014). *Cross-domain Recommendations based on semantically-enhanced User Web Behavior*. PhD thesis, Karlsruhe, Karlsruher Institut für Technologie (KIT), Diss., 2014.

[Huang et al., 2010] Huang, H.-D., Chuang, T.-Y., Tsai, Y.-L., and Lee, C.-S. (2010). Ontology-based intelligent system for malware behavioral analysis. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–6. IEEE.

[Isele et al., 2010] Isele, R., Jentzsch, A., and Bizer, C. (2010). Silk server-adding missing links while consuming linked data. *COLD*, 665.

[Ishaq et al., 2012] Ishaq, I., Hoebeke, J., Rossey, J., De Poorter, E., Moerman, I., and Demeester, P. (2012). Facilitating sensor deployment, discovery and resource access using embedded web services. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 717–724. IEEE.

[Janowicz et al., 2014] Janowicz, K., Hitzler, P., Adams, B., Kolas, D., and Vardeman II, C. (2014). Five stars of linked data vocabulary use. *Semantic Web*, 5(3):173–176.

[Jara et al., 2014] Jara, A. J., Olivieri, A. C., Bocchi, Y., Jung, M., Kastner, W., and Skarmeta, A. F. (2014). Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. *International Journal of Web and Grid Services*, 10(2):244–272.

[Jennings et al., 2012] Jennings, C., Arkko, J., and Shelby, Z. (2012). Media types for sensor markup language (senml).

[Jentzsch et al., 2010] Jentzsch, A., Isele, R., and Bizer, C. (2010). Silk–generating rdf links while publishing or consuming linked data. In *9th International Semantic Web Conference (ISWC10)*. Citeseer.

[Jeong et al., 2006] Jeong, K., Choi, D., Kim, S., and Lee, G. (2006). A middleware architecture determining application context using shared ontology. *Computational Science and Its Applications-ICCSA 2006*, pages 128–137.

[Jiménez-Ruiz and Grau, 2011] Jiménez-Ruiz, E. and Grau, B. C. (2011). Logmap: Logic-based and scalable ontology matching. In *The Semantic Web–ISWC 2011*, pages 273–288. Springer.

[Joshi, 2013] Joshi, A. P. (2013). *Linked Data for Software Security Concepts and Vulnerability Descriptions*. PhD thesis, University of Maryland.

[Jung et al., 2012] Jung, S.-J., Myllyla, R., and Chung, W.-Y. (2012). A wireless machine-to-machine healthcare solution using android mobile devices in global networks.

[Jurdak et al., 2004] Jurdak, R., Lopes, C., and Baldi, P. (2004). A framework for modeling sensor networks. In *Proceedings of the Building Software for Pervasive Computing Workshop at OOPSLA*, volume 4, pages 1–5.

[Kalem and Turhan, 2005] Kalem, G. and Turhan, Ç. (2005). Semantic web application: Ontology-driven recipe querying. Master's thesis.

[Kalfoglou and Schorlemmer, 2003] Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The knowledge engineering review*, 18(01):1–31.

[Karlof et al., 2004] Karlof, C., Sastry, N., and Wagner, D. (2004). Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175. ACM.

[Kazakov et al., 2012] Kazakov, Y., Krötzsch, M., and Simancik, F. (2012). Elk reasoner: Architecture and evaluation. In *ORE*.

[Kenfack et al., 2011] Kenfack, H., Ndié, T., Nataf, E., Festor, O., et al. (2011). Une ontologie pour la description des intrusions dans les rcsfs. In *CFIP 2011-Colloque Francophone sur l Ingénierie des Protocoles*.

[Khandelwal et al., 2011] Khandelwal, A., Jacobi, I., and Kagal, L. (2011). Linked rules: principles for rule reuse on the web. In *Web Reasoning and Rule Systems*, pages 108–123. Springer.

[Khriyenko et al., 2012] Khriyenko, O., Terziyan, V., and Kaikova, O. (2012). User-assisted semantic interoperability in internet of things: Visually-facilitated ontology alignment through visually-enriched ontology and thing descriptions. In *UBICOMM 2012, The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 104–110.

[Khusro et al., 2013] Khusro, S., Ali, S., Rauf, A., Mahfooz, S., Mashwani, S. R., Khan, A., and Jabeen, F. (2013). Unleashing sensor data on linked open data-the story so far. *Life Science Journal*, 10(4).

[Kifer, 2008] Kifer, M. (2008). Rule interchange format: The framework. In *Web reasoning and rule systems*, pages 1–11. Springer.

[Kim et al., 2005] Kim, A., Luo, J., and Kang, M. (2005). Security ontology for annotating resources. *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, pages 1483–1499.

[Kim et al., 2008] Kim, J., Kwon, H., Kim, D., Kwak, H., and Lee, S. (2008). 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.

[Kitsos et al., 2007] Kitsos, P., Sklavos, N., and Koufopavlou, O. (2007). Umts security: system architecture and hardware implementation. *Wireless Communications and Mobile Computing*, 7(4):483–494.

[Kofler et al., 2011] Kofler, M., Reinisch, C., and Kastner, W. (2011). An intelligent knowledge representation of smart home energy parameters. In *Proceedings of the World Renewable Energy Congress (WREC 2011), Linköping, Sweden*.

[Kofler et al., 2012] Kofler, M. J., Reinisch, C., and Kastner, W. (2012). A semantic representation of energy-related information in future smart homes. *Energy and Buildings*, 47:169–179.

[Koien, 2004] Koien, G. M. (2004). An introduction to access security in umts. *Wireless Communications, IEEE*, 11(1):8–18.

[Kolozali et al., 2014] Kolozali, S., Elsaleh, T., and Barnaghi, P. (2014). A validation tool for the w3c ssn ontology based sensory semantic knowledge.

[Kuntarto and Gunawan, 2012] Kuntarto, G. P. and Gunawan, D. (2012). Dwipa search engine: When e-tourism meets the semantic web. In *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*, pages 155–160. IEEE.

[Lassila and Swick, 1999] Lassila, O. and Swick, R. R. (1999). Resource description framework (rdf) model and syntax specification. http://www.w3.org/TR/REC-rdf-syntax/.

[Le Phuoc, 2013] Le Phuoc, D. (2013). *A native and adaptive approach for linked stream data processing*. PhD thesis.

[Le-Phuoc et al., 2011] Le-Phuoc, D., Dao-Tran, M., Parreira, J. X., and Hauswirth, M. (2011). A native and adaptive approach for unified processing of linked streams and linked data. In *The Semantic Web–ISWC 2011*, pages 370–388. Springer.

[Le Phuoc et al., 2010] Le Phuoc, D., Parreira, J. X., Reynolds, V., and Hauswirth, M. (2010). Rdf on the go: Rdf storage and query processor for mobile devices. In *ISWC Posters&Demos*, page 12.

[Le-Phuoc et al., 2014] Le-Phuoc, D., Quoc, H. N. M., Ngo, Q. H., Nhat, T. T., and Hauswirth, M. (2014). Enabling live exploration on the graph of things? *Proceedings of the Semantic Web Challenge*.

[Lécué et al., 2012] Lécué, F., Schumann, A., and Sbodio, M. L. (2012). Applying semantic web technologies for diagnosing road traffic congestions. In *The Semantic Web–ISWC 2012*, pages 114–130. Springer.

[Lécué et al., 2014a] Lécué, F., Tallevi-Diotallevi, S., Hayes, J., Tucker, R., Bicer, V., Sbodio, M. L., and Tommasi, P. (2014a). Star-city: semantic traffic analytics and reasoning for city. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 179–188. ACM.

[Lécué et al., 2014b] Lécué, F., Tucker, R., Bicer, V., Tommasi, P., Tallevi-Diotallevi, S., and Sbodio, M. (2014b). Predicting severity of road traffic congestion using semantic web technologies. In *The Semantic Web: Trends and Challenges*, pages 611–627. Springer.

[Lécué et al., 2014c] Lécué, F., Tucker, R., Tallevi-Diotallevi, S., Nair, R., Gkoufas, Y., Liguori, G., Borioni, M., Rademaker, A., and Barbosa, L. (2014c). Semantic traffic diagnosis with star-city: Architecture and lessons learned from deployment in dublin, bologna, miami and rio. In *The Semantic Web–ISWC 2014*, pages 292–307. Springer.

[Lee et al., 2013] Lee, G. M., Crespi, N., Choi, J. K., and Boussard, M. (2013). Internet of things. In *Evolution of Telecommunication Services*, pages 257–282. Springer.

[Leggieri et al., 2011] Leggieri, M., Passant, A., and Hauswirth, M. (2011). incontext-sensing: Lod augmented sensor data? In *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*. Citeseer.

[Lekhchine, 2009] Lekhchine, R. (2009). *Construction d'une ontologie pour le domaine de la securite : application aux agents mobiles*. PhD thesis, University Mentouri.

[Lighfoot et al., 2007] Lighfoot, L. E., Ren, J., and Li, T. (2007). An energy efficient link-layer security protocol for wireless sensor networks. In *Electro/Information Technology, 2007 IEEE International Conference on*, pages 233–238. IEEE.

[Lindstrom and Jeffries, 2004] Lindstrom, L. and Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information Systems Management*, 21(3):41–52.

[Liu, 2013] Liu, H. (2013). Improving semantic interoperability in remote patient monitoring. Master's thesis.

[Logre et al., 2014] Logre, I., Mosser, S., Collet, P., and Riveill, M. (2014). Sensor data visualisation: a composition-based approach to support domain variability. pages 1–16.

[López et al., 2008] López, J., Gil, R., García, R., Cearreta, I., and Garay, N. (2008). Towards an ontology for describing emotions. *Emerging Technologies and Information Systems for the Knowledge Society*, pages 96–104.

[Luk et al., 2007] Luk, M., Mezzour, G., Perrig, A., and Gligor, V. (2007). Minisec: a secure sensor network communication architecture. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 479–488. ACM.

[M2M, 2012] M2M, E. (2012). Machine-to-Machine Communications (M2M); Study on Semantic support for M2M data, ETSI Techinal Report 101 584 v2.1.1 (2013-12).

[Maedche et al., 2002] Maedche, A., Motik, B., Silva, N., and Volz, R. (2002). Mafraa mapping framework for distributed ontologies. In *Knowledge engineering and knowledge management: ontologies and the semantic web*, pages 235–250. Springer.

[Malan et al., 2004] Malan, D., Fulford-Jones, T., Welsh, M., and Moulton, S. (2004). Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International workshop on wearable and implantable body sensor networks*, volume 5.

[Manate et al., 2014] Manate, B., Munteanu, V. I., and Fortis, T. F. (2014). Towards a smarter internet of things: Semantic visions. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on*, pages 582–587.

[Mandler et al., 2013] Mandler, B., Antonelli, F., Kleinfeld, R., Pedrinaci, C., Carrera, D., Gugliotta, A., Schreckling, D., Carreras, I., Raggett, D., Pous, M., et al. (2013). Compose–a journey from the internet of things to the internet of services. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 1217–1222. IEEE.

[Maurer and Martel, 2002] Maurer, F. and Martel, S. (2002). Extreme programming: Rapid development for web-based applications. *IEEE Internet computing*, 6(1):86–90.

[McBride, 2002] McBride, B. (2002). Jena: A semantic web toolkit. *Internet Computing, IEEE*, 6(6):55–59.

[McIlraith et al., 2001] McIlraith, S. A., Son, T. C., and Zeng, H. (2001). Semantic web services. *IEEE intelligent systems*, 16(2):46–53.

[Meyer and Wetzel, 2004] Meyer, U. and Wetzel, S. (2004). A man-in-the-middle attack on umts. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97. ACM.

[Miao et al., 2013] Miao, L., Chun, J., and Yoshiyuki HIGUCHI, J. C. (2013). Ontology-based user preferences bayesian model for personalized recommendation. *Journal of Computational Information Systems*, 9(16):6579–6586.

[Mietz et al., 2013] Mietz, R., Groppe, S., Römer, K., and Pfisterer, D. (2013). Semantic models for scalable search in the internet of things. *Journal of Sensor and Actuator Networks*, 2(2):172–195.

[Miorandi et al., 2012] Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516.

[Mobarhan et al., 2012] Mobarhan, M. A., Mobarhan, M. A., and Shahbahrami, A. (2012). Evaluation of security attacks on umts authentication mechanism. *International Journal of Network Security & Its Applications*, 4(4).

[Moraru, 2011] Moraru, A. (2011). Enrichment of sensor descriptions and measuremebts using semantic technologies. Master's thesis.

[Moraru and Mladenić, 2012] Moraru, A. and Mladenić, D. (2012). A framework for semantic enrichment of sensor data. *CIT. Journal of Computing and Information Technology*, 20(3):167–173.

[Moraru et al., 2011] Moraru, A., Mladenic, D., Vucnik, M., Porcius, M., Fortuna, C., and Mohorcic, M. (2011). Exposing real world information for the web of things. In *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*, page 6. ACM.

[Moraru et al., 2010] Moraru, A., Pesko, M., Porcius, M., Fortuna, C., and Mladenic, D. (2010). Using machine learning on sensor data. *CIT. Journal of Computing and Information Technology*, 18(4):341–347.

[Morignot and Nashashibi, 2012] Morignot, P. and Nashashibi, F. (2012). An ontology-based approach to relax traffic regulation for autonomous vehicle assistance. *CoRR*, abs/1212.0768.

[Mosser et al., 2013] Mosser, S., Logre, I., Ferry, N., and Collet, P. (2013). From Sensors to Visualization Dashboards: Need for Language Composition. In *Globalization of Modelling Languages workshop(GeMOC'13), Miami, 29 September 2013*, pages 1–6. IEEE.

[Motik et al., 2012] Motik, B., Horrocks, I., and Kim, S. M. (2012). Delta-reasoner: a semantic web reasoner for an intelligent mobile platform. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 63–72. ACM.

[Nambi et al., 2014] Nambi, S., Sarkar, C., Prasad, R. V., and Rahim, A. (2014). A unified semantic knowledge base for iot. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 575–580. IEEE.

[Neji and Bouallegue, 2012a] Neji, H. and Bouallegue, R. (2012a). Roadmap for establishing interoperability of heterogeneous cellular network technologies-1. *arXiv preprint arXiv:1207.3358*.

[Neji and Bouallegue, 2012b] Neji, H. and Bouallegue, R. (2012b). Roadmap for establishing interoperability of heterogeneous cellular network technologies-2. *Journal of Signal and Information Processing*, 3(3):402–411.

[Neji and Bouallegue, 2012c] Neji, H. and Bouallegue, R. (2012c). Roadmap for establishing interoperability of heterogeneous cellular network technologies-3. *International Journal of Computer Applications*, 54(5):17–27.

[Nijdam, 2009] Nijdam, N. A. (2009). Mapping emotion to color.

[Noy et al., 2001] Noy, N. F., McGuinness, D. L., et al. (2001). Ontology development 101: A guide to creating your first ontology.

[Noy and Musen, 2001] Noy, N. F. and Musen, M. A. (2001). Anchor-prompt: Using non-local context for semantic matching. In *Proceedings of the workshop on ontologies and information sharing at the international joint conference on artificial intelligence (IJCAI)*, pages 63–70.

[Okeyo et al., 2013] Okeyo, G., Chen, L., and Wang, H. (2013). An agent-mediated ontology-based approach for composite activity recognition in smart homes. *J. UCS*, 19(17):2577–2597.

[OneM2M et al., 2014] OneM2M, Abstraction, W. M., and Semantics (2014). oneM2M Technical Report 0007 Study of Abstraction and Semantics Enablement v.0.7.0, Study of Existing Abstraction and Semantic Capability Enablement Technologies for consideration by oneM2M.

[OneM2M and Requirements, 2014] OneM2M and Requirements, W. (2014). oneM2M Technical Report 0001 Use Cases.

[OneM2M and Security, 2014] OneM2M and Security, W. (2014). oneM2M Technical Report 0008 Security v.1.0.0, Security Issues from Use Cases.

[Oren et al., 2009] Oren, E., Kotoulas, S., Anadiotis, G., Siebes, R., ten Teije, A., and van Harmelen, F. (2009). Marvin: Distributed reasoning over large-scale semantic web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):305–316.

[Öspinar, 2014] Öspinar, M. (2014). *A Flexible Semantic Service Composition Framework for Pervasive Computing Environments*. PhD thesis, Moddle East Technical University.

[Paganelli et al., 2014] Paganelli, F., Turchi, S., and Giuli, D. (2014). A web of things framework for restful applications and its experimentation in a smart city.

[Park et al., 2014] Park, K., Kim, Y., and Chang, J. (2014). Semantic reasoning with contextual ontologies on sensor cloud environment. *International Journal of Distributed Sensor Networks*, 2014.

[Park and Park, 2007] Park, Y. and Park, T. (2007). A survey of security threats on 4g networks. In *Globecom Workshops, 2007 IEEE*, pages 1–6. IEEE.

[Patel et al., 2013] Patel, P., Pathak, A., Cassou, D., and Issarny, V. (2013). Enabling high-level application development in the internet of things. In Zuniga, M. and Dini, G., editors, *Sensor Systems and Software*, volume 122 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 111–126. Springer International Publishing.

[Patel et al., 2011] Patel, P., Pathak, A., Teixeira, T., and Issarny, V. (2011). Towards application development for the internet of things. In *Proceedings of the 8th Middleware Doctoral Symposium*, MDS '11, pages 5:1–5:6, New York, NY, USA. ACM.

[Patkos et al., 2010] Patkos, T., Chrysakis, I., Bikakis, A., Plexousakis, D., and Antoniou, G. (2010). A reasoning framework for ambient intelligence. In *Artificial Intelligence: Theories, Models and Applications*, pages 213–222. Springer.

[Patni et al., 2010a] Patni, H., Henson, C., and Sheth, A. (2010a). Linked sensor data. In *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pages 362–370. IEEE.

[Patni et al., 2010b] Patni, P., Sahoo, S., Henson, C., and Sheth, A. (2010b). Provenance aware linked sensor data. In *Proceedings of the Second Workshop on Trust and Privacy on the Social and Semantic Web*.

[Perera et al., 2014] Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *Communications Surveys Tutorials, IEEE*, 16(1):414–454.

[Perrig et al., 2002] Perrig, A., Szewczyk, R., Tygar, J., Wen, V., and Culler, D. E. (2002). Spins: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534.

[Pfisterer et al., 2011] Pfisterer, D., Romer, K., Bimschas, D., Kleine, O., Mietz, R., Truong, C., Hasemann, H., Kroller, A., Pagel, M., Hauswirth, M., et al. (2011). Spitfire: toward a semantic web of things. *Communications Magazine, IEEE*, 49(11):40–48.

[Phuoc and Hauswirth, 2009] Phuoc, D. and Hauswirth, M. (2009). Linked open data in sensor data mashups.

[Pizzuti and Mirabelli, 2013] Pizzuti, T. and Mirabelli, G. (2013). Ftto: An example of food ontology for traceability purpose. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on*, volume 1, pages 281–286. IEEE.

[Pollard et al., 2013] Pollard, E., Morignot, P., and Nashashibi, F. (2013). An Ontology-based Model to Determine the Automation Level of an Automated Vehicle for Co-Driving. In *16th International Conference on Information Fusion*, Istanbul, Turquie.

[Poveda-Villalón et al., 2012a] Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2012a). Did you validate your ontology? oops!

[Poveda-Villalón et al., 2012b] Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2012b). Validating ontologies with oops! In *Knowledge Engineering and Knowledge Management*, pages 267–281. Springer.

[Preuveneers et al., 2004] Preuveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., and De Bosschere, K. (2004). Towards an extensible context ontology for ambient intelligence. In *Ambient intelligence*, pages 148–159. Springer.

[Prud'Hommeaux et al., 2006] Prud'Hommeaux, E., Seaborne, A., et al. (2006). Sparql query language for rdf. http://www.w3.org/TR/rdf-sparql-query/.

[Pschorr et al., 2010] Pschorr, J., Henson, C., Patni, H., and Sheth, A. (2010). Sensor discovery on linked data. In *Proceedings of the 7th Extended Semantic Web Conference, ESWC2010, Heraklion, Greece*, volume 30.

[Qin et al., 2014] Qin, Y., Sheng, Q. Z., Falkner, N. J., Dustdar, S., and Wang, H. (2014). When things matter: A data-centric view of the internet of things. *arXiv preprint arXiv:1407.2704*.

[Rahman and Sharma, 2012] Rahman, A. and Sharma, K. (2012). Fourth generation of mobile communication network: Evolution, prospects, objectives, challenges and security.

[Ramparany et al., 2014] Ramparany, F., Marquez, F. G., Soriano, J., and Elsaleh, T. (2014). Handling smart environment devices, data and services at the semantic level with the fi-ware core platform. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 14–20. IEEE.

[Rana, 2011] Rana, P. (2011). Securing 4 g networks with y-communication using aka protocol.

[Ranganathan et al., 2003] Ranganathan, A., McGrath, R. E., Campbell, R. H., and Mickunas, M. D. (2003). Use of ontologies in a pervasive computing environment. *The Knowledge Engineering Review*, 18(03):209–220.

[Raul Garcia-Castro, 2014] Raul Garcia-Castro, Asuncion Gomez-Perez, O. C. (2014). Ready4smartcities: Ict roadmap and data interoperability for energy systems in smart cities. In *2014 European Semantic Web Conference*.

[Razzaq et al., 2014] Razzaq, A., Latif, K., Ahmad, H. F., Hur, A., Anwar, Z., and Bloodsworth, P. C. (2014). Semantic security against web application attacks. *Information Sciences*, 254:19–38.

[Rector et al., 2004] Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., and Wroe, C. (2004). Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *Engineering Knowledge in the Age of the Semantic Web*, pages 63–81. Springer.

[Reinisch et al., 2011] Reinisch, C., Kofler, M., Iglesias, F., and Kastner, W. (2011). Thinkhome energy efficiency in future smart homes. *EURASIP Journal on Embedded Systems*, 2011:1.

[Riboni and Bettini, 2011] Riboni, D. and Bettini, C. (2011). Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289.

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

[Rodríguez et al., 2013] Rodríguez, J. M. Á., Gayo, J. E. L., and De Pablos, P. O. (2013). Ontospread: A framework for supporting the activation of concepts in graph-based structures through the spreading activation technique. In *Information Systems, E-learning, and Knowledge Management Research*, pages 454–459. Springer.

[Rodríguez et al., 2014] Rodríguez, N. D., Cuéllar, M. P., Lilius, J., and Calvo-Flores, M. D. (2014). A survey on ontologies for human behavior recognition. *ACM Computing Surveys (CSUR)*, 46(4):43.

[Rodríguez-Molina et al., 2013] Rodríguez-Molina, J., Martínez, J.-F., Castillejo, P., and López, L. (2013). Combining wireless sensor networks and semantic middleware for an internet of things-based sportsman/woman monitoring application. *Sensors*, 13(2):1787–1835.

[Russomanno et al., 2005a] Russomanno, D., Kothari, C., and Thomas, O. (2005a). Sensor ontologies: from shallow to deep models. In *System Theory, 2005. SSST'05. Proceedings of the Thirty-Seventh Southeastern Symposium on*, pages 107–112. IEEE.

[Russomanno et al., 2005b] Russomanno, D. J., Kothari, C., and Thomas, O. (2005b). Building a sensor ontology: A practical approach leveraging iso and ogc models. In *The 2005 International Conference on Artificial Intelligence*, volume 201.

[Ruta and Scioscia, 2014] Ruta, M. and Scioscia, F. (2014). Framework and architecture for the semantic web of things.

[Ruta et al., 2012] Ruta, M., Scioscia, F., and Di Sciascio, E. (2012). Enabling the semantic web of things: Framework and architecture. In *ICSC*, pages 345–347.

[Ruta et al., 2010] Ruta, M., Scioscia, F., Gramegna, F., and Di Sciascio, E. (2010). A mobile knowledge-based system for on-board diagnostics and car driving assistance. In *UBICOMM 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 91–96.

[Salahi and Ansarinia, 2013] Salahi, A. and Ansarinia, M. (2013). Predicting network attacks using ontology-driven inference. *arXiv preprint arXiv:1304.0913*.

[Sanchez et al., 2011] Sanchez, L., Galache, J. A., Gutierrez, V., Hernández, J. M., Bernat, J., Gluhak, A., and García, T. (2011). Smartsantander: The meeting point between future internet research and experimentation and the smart cities. In *Future Network & Mobile Summit (FutureNetw), 2011*, pages 1–8. IEEE.

[Sanchez et al., 2014] Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., et al. (2014). Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238.

[Scharffe et al., 2012] Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Képéklian, G., Cotton, F., et al. (2012). Enabling linked data publication with the datalift platform. In *Proc. AAAI workshop on semantic cities*, pages No–pagination.

[Scharffe and Euzenat, 2011] Scharffe, F. and Euzenat, J. (2011). Melinda: an interlinking framework for the web of data. *arXiv preprint arXiv:1107.4502*.

[Schlicht and Stuckenschmidt, 2010] Schlicht, A. and Stuckenschmidt, H. (2010). Peer-to-peer reasoning for interlinked ontologies. *International Journal of Semantic Computing*, 4(01):27–58.

[Seddigh et al., 2010] Seddigh, N., Nandy, B., Makkar, R., and Beaumont, J.-F. (2010). Security advances and challenges in 4g wireless networks. In *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*, pages 62–71. IEEE.

[Sequeda and Corcho, 2009] Sequeda, J. F. and Corcho, O. (2009). Linked stream data: A position paper.

[Serafini and Tamilin, 2005] Serafini, L. and Tamilin, A. (2005). Drago: Distributed reasoning architecture for the semantic web. In *The Semantic Web: Research and Applications*, pages 361–376. Springer.

[Serrano et al., 2013] Serrano, M., Quoc, H. N. M., Hauswirth, M., Wang, W., Barnaghi, P., and Cousin, P. (2013). Open services for iot cloud applications in the future internet. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE.

[Seye et al., 2012] Seye, O., Faron-Zucker, C., Corby, O., and Follenfant, C. (2012). Bridging the gap between rif and sparql: Implementation of a rif dialect with a sparql rule engine. *AImWD 2012*, page 19.

[Sharma et al., 2012] Sharma, M., Sharma, D., and Khan, S. (2012). Honey as complementary medicine:-a review. *International Journal of Pharma and Bio Sciences*.

[Shelby et al., 2012] Shelby, Z., Hartke, K., Bormann, C., and Frank, B. (2012). Constrained application protocol (coap), draft-ietf-core-coap-13. *Orlando: The Internet Engineering Task Force–IETF, Dec*.

[Sheth et al., 2013] Sheth, A., Anantharam, P., and Henson, C. (2013). Physical-cyber-social computing: An early 21st century approach. *Intelligent Systems, IEEE*, 28(1):78–82.

[Sheth et al., 2008] Sheth, A., Henson, C., and Sahoo, S. (2008). Semantic sensor web. *Internet Computing, IEEE*, 12(4):78–83.

[Shvaiko and Euzenat, 2013] Shvaiko, P. and Euzenat, J. (2013). Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):158–176.

[Simperl, 2009] Simperl, E. (2009). Reusing ontologies on the semantic web: A feasibility study. *Data & Knowledge Engineering*, 68(10):905–925.

[Sivieri et al., 2014] Sivieri, A., Cugola, G., Baresi, L., and Fiorini, C. (2014). Eliot: A programming framework for the internet of things.

[Skillen et al., 2012] Skillen, K., Chen, L., Nugent, C. D., Donnelly, M. P., and Solheim, I. (2012). A user profile ontology based approach for assisting people with dementia in mobile environments. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 6390–6393. IEEE.

[Skillen et al., 2013] Skillen, K.-L., Chen, L., Nugent, C. D., Donnelly, M. P., Burns, W., and Solheim, I. (2013). Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments. *Future Generation Computer Systems*.

[Sorrentino, 2012] Sorrentino, D. (2012). Integrating semantic networks and object-oriented model to represent and manage context. Master's thesis.

[Souag et al., 2012] Souag, A., Salinesi, C., and Comyn-Wattiau, I. (2012). Ontologies for security requirements: A literature survey and classification. In *Advanced Information Systems Engineering Workshops*, pages 61–69. Springer.

[Staroch, 2013] Staroch, P. (2013). A weather ontology for predictive control in smart homes. Master's thesis.

[Stavropoulos et al., 2012] Stavropoulos, T. G., Vrakas, D., Vlachava, D., and Bassiliades, N. (2012). Bonsai: a smart building ontology for ambient intelligence. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 30. ACM.

[Stecher et al., 2008] Stecher, R., Niederée, C., Nejdl, W., and Bouquet, P. (2008). Adaptive ontology re-use: finding and re-using sub-ontologies. *International Journal of Web Information Systems*, 4(2):198–214.

[Stevenson et al., 2009] Stevenson, G., Knox, S., Dobson, S., and Nixon, P. (2009). Ontonym: a collection of upper ontologies for developing pervasive systems. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, page 9. ACM.

[Stocker et al., 2012] Stocker, M., Rönkkö, M., and Kolehmainen, M. (2012). *Making sense of sensor data using ontology: A discussion for road vehicle classification*. PhD thesis, International Environmental Modelling and Software Society (iEMSs).

[Su et al., 2012] Su, C. J., Chih, C. W., Chen, Y. A., et al. (2012). Ontology-based personalized diet plan web service using hl7 health screening data.

[Su et al., 2014a] Su, X., Riekki, J., Nurminen, J. K., Nieminen, J., and Koskimies, M. (2014a). Adding semantics to internet of things. *Concurrency and Computation: Practice and Experience*.

[Su et al., 2014b] Su, X., Zhang, H., Riekki, J., Keränen, A., Nurminen, J. K., and Du, L. (2014b). Connecting iot sensors to knowledge-based systems by transforming senml to rdf. *Procedia Computer Science*, 32:215–222.

[Suárez-Figueroa, 2010] Suárez-Figueroa, M. C. (2010). *NeOn Methodology for building ontology networks: specification, scheduling and reuse*. PhD thesis, Informatica.

[Suarez-Figueroa et al., 2012] Suarez-Figueroa, M. C., Gomez-Perez, A., and Fernandez-Lopez, M. (2012). The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer.

[Suksom et al., 2010] Suksom, N., Buranarach, M., Thein, Y. M., Supnithi, T., and Netisopakul, P. (2010). A knowledge-based framework for development of personalized food recommender system. In *Proc. of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems*.

[Swetina et al., 2014] Swetina, J., Lu, G., Jacobs, P., Ennesser, F., and Song, J. (2014). Toward a standardized common m2m service layer platform: Introduction to onem2m. *Wireless Communications, IEEE*, 21(3):20–26.

[Taylor et al., 2013] Taylor, K., Lamb, D., Griffith, C., Falzon, G., Lefort, L., Gaire, R., Compton, M., Trotter, M., and Wark, T. (2013). Farming the web of things.

[Tobías, 2013] Tobías, I. F. (2013). *A semantic-based framework for building cross-domain networks: Application to item recommendation*. PhD thesis.

[Truong et al., 2011] Truong, H. B., Nguyen, N. T., and Nguyen, P. K. (2011). Fuzzy ontology building and integration for fuzzy inference systems in weather forecast domain. In *Intelligent Information and Database Systems*, pages 517–527. Springer.

[Tsoumas et al., 2005] Tsoumas, B., Dritsas, S., and Gritzalis, D. (2005). An ontology-based approach to information systems security management. *Computer Network Security*, pages 151–164.

[Tu, 2009] Tu, S. (2009). Exploiting linked data to build web applications.

[Tummarello et al., 2007] Tummarello, G., Delbru, R., and Oren, E. (2007). *Sindice.com: Weaving the open linked data*. Springer.

[Tumnark et al., 2013] Tumnark, P., Oliveira, L., and Santibutr, N. (2013). Ontology-based personalized dietary recommendation for weightlifting. In *2013 International Workshop on Computer Science in Sports*. Atlantis Press.

[Undercoffer et al., 2003] Undercoffer, J., Joshi, A., and Pinkston, J. (2003). Modeling computer attacks: An ontology for intrusion detection. In *Recent Advances in Intrusion Detection*, pages 113–135. Springer.

[Urbani, 2013] Urbani, J. (2013). *On web-scale reasoning*. PhD thesis.

[Urbani et al., 2012] Urbani, J., Kotoulas, S., Maassen, J., Van Harmelen, F., and Bal, H. (2012). Webpie: A web-scale parallel inference engine using mapreduce. *Web Semantics: Science, Services and Agents on the World Wide Web*, 10:59–75.

[Vandenbussche et al., 2015] Vandenbussche, P.-Y., Atemezing, G. A., Poveda-Villalónc, M., and Vatantd, B. (2015). Lov: a gateway to reusable semantic vocabularies on the web. *Semantic Web Journal*.

[Vandenbussche and Vatant, 2011] Vandenbussche, P.-Y. and Vatant, B. (2011). Metadata recommendations for linked open data vocabularies. *Version*, 1:2011–12.

[Vandenbussche et al., 2012] Vandenbussche, P.-Y., Vatant, B., and Charlet, J. (2012). Linked open vocabularies, un écosystème encore fragile. *White paper, Mondeca*.

[Vasileios and Antoniou, 2012] Vasileios, E. and Antoniou, G. (2012). A real-time semantics-aware activity recognition system.

[Villata et al., 2011a] Villata, S., Delaforge, N., Gandon, F., and Gyrard, A. (2011a). An access control model for linked data. In *On the Move to Meaningful Internet Systems: OTM 2011 Workshops*, pages 454–463. Springer.

[Villata et al., 2011b] Villata, S., Delaforge, N., Gandon, F., and Gyrard, A. (2011b). Social semantic web access control. *Proceedings of the 4th international workshop social data on the web (SDoW-2011)*, pages 48–59.

[Vincent et al., 2012] Vincent, J., Dubin, T., Porquet, C., et al. (2012). Protection de la vie privée basée sur des ontologies dans un système android. *APVP 2012 (Atelier Protection de la Vie Privée, 3ème édition)*.

[Vincent et al., 2011a] Vincent, J., Porquet, C., Borsali, M., and Leboulanger, H. (2011a). Privacy protection for smartphones: an ontology-based firewall. In *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*, pages 371–380. Springer.

[Vincent et al., 2011b] Vincent, J., Porquet, C., Oulmakhzoune, I., et al. (2011b). Ontology-based privacy protection for smartphone: A firewall implementation. In *International Conference on Secure Networking and Applications (ICSNA)*.

[Volz et al., 2009a] Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. (2009a). *Discovering and maintaining links on the web of data*. Springer.

[Volz et al., 2009b] Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. (2009b). Silk–a link discovery framework for the web of data. In *Proceedings of the 2nd Linked Data on the Web Workshop*, pages 559–572. Citeseer.

[Vorobiev and Bekmamedova, 2010] Vorobiev, A. and Bekmamedova, N. (2010). An ontology-driven approach applied to information security. *Journal of Research and Practice in Information Technology*, 42(1):61.

[Wang et al., 2013] Wang, W., De, S., Cassar, G., and Moessner, K. (2013). Knowledge representation in the internet of things: semantic modelling and its applications. *Automatika–Journal for Control, Measurement, Electronics, Computing and Communications*, 54(4).

[Wang et al., 2008] Wang, W., Zeng, G., Zhang, D., Huang, Y., Qiu, Y., and Wang, X. (2008). An intelligent ontology and bayesian network based semantic mashup for tourism. In *Services-Part I, 2008. IEEE Congress on*, pages 128–135. IEEE.

[Wang et al., 2002] Wang, X., Dong, J. S., Chin, C., Hettiarachchi, S. R., and Zhang, D. (2002). Semantic space: An infrastructure for smart spaces. *Computing*, 1(2):67–74.

[Wang et al., 2004] Wang, X. H., Zhang, D. Q., Gu, T., and Pung, H. K. (2004). Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22. IEEE.

[Wang et al., 2010] Wang, Y., Wang, S., Stash, N., Aroyo, L., and Schreiber, G. (2010). Enhancing content-based recommendation with the task model of classification. In *Knowledge Engineering and Management by the Masses*, pages 431–440. Springer.

[Wei and Barnaghi, 2009] Wei, W. and Barnaghi, P. (2009). Semantic annotation and reasoning for sensor data. *Smart Sensing and Context*, pages 66–76.

[Weiser, 1993] Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84.

[Welty et al., 2004] Welty, C., McGuinness, D. L., and Smith, M. K. (2004). Owl web ontology language guide. *W3C recommendation, W3C (February 2004) http://www. w3. org/TR/2004/REC-owl-guide-20040210*.

[Wongpatikaseree et al., 2012] Wongpatikaseree, K., Ikeda, M., Buranarach, M., Supnithi, T., Lim, A. O., and Tan, Y. (2012). Activity recognition using context-aware infrastructure ontology in smart home domain. In *Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on*, pages 50–57. IEEE.

[Xenakis and Merakos, 2004] Xenakis, C. and Merakos, L. (2004). Security in third generation mobile networks. *Computer communications*, 27(7):638–650.

[Xie et al., 2013] Xie, L., Yin, Y., Lu, X., Sheng, B., and Lu, S. (2013). ifridge: An intelligent fridge for food management based on rfid technology. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 291–294. ACM.

[Xu et al., 2014] Xu, L., He, W., and Li, S. (2014). Internet of things in industries: A survey. *Industrial Informatics, IEEE Transactions on*, PP(99):1–1.

[Ye et al., 2015] Ye, J., Dasiopoulou, S., Stevenson, G., Meditskos, G., Kontopoulos, E., Kompatsiaris, I., and Dobson, S. (2015). Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing*.

[Ye et al., 2011] Ye, J., Stevenson, G., and Dobson, S. (2011). A top-level ontology for smart environments. *Pervasive and mobile computing*, 7(3):359–378.

[Zeng et al., 2011] Zeng, D., Guo, S., and Cheng, Z. (2011). The web of things: A survey. *Journal of Communications*, 6(6):424–438.

[Zhang et al., 2005] Zhang, D., Gu, T., and Wang, X. (2005). Enabling context-aware smart home with semantic web technologies. *International Journal of Human-friendly Welfare Robotic Systems*, 6(4):12–20.

[Zhang et al., 2011] Zhang, Y., Yu, R., Xie, S., Yao, W., Xiao, Y., and Guizani, M. (2011). Home m2m networks: Architectures, standards, and qos improvement. *Communications Magazine, IEEE*, 49(4):44–52.

[Zhu et al., 2006] Zhu, S., Setia, S., and Jajodia, S. (2006). Leap+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(4):500–528.

[Ziafati et al., 2011] Ziafati, P., Mastrogiovanni, F., and Sgorbissa, A. (2011). Fast prototyping and deployment of context-aware smart outdoor environments. In *Intelligent Environments (IE), 2011 7th International Conference on*, pages 206–213. IEEE.

[Znaidi et al., 2008] Znaidi, W., Minier, M., Babau, J., et al. (2008). An ontology for attacks in wireless sensor networks.

[Zografistou, 2012] Zografistou, D. (2012). Support for context-aware healthcare in ambient assisted living. Master's thesis.

Appendix A

List of Publications

"Do not judge me by my successes, judge me by how many times I fell down and got back up again."

Nelson Mandela

This thesis and its contributions are based on several publications or disseminations of our work in standardizations which have been exploited for writing this manuscript.

A.1 International Conferences

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud. *"Enrich machine-to-machine data with semantic web technologies for cross-domain applications"* In WF-IOT 2014, World Forum on Internet of Things, 6-8 March 2014, Seoul, Korea
- Amelie Gyrard, Christian Bonnet and Karima Boudaoud *"An ontology-based approach for helping to secure the ETSI Machine-to-Machine Architecture"* IEEE International Conference on Internet of Things 2014 (iThings), 1-3 September 2014, Taipei, Taiwan

A.2 International Workshops

- Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, Karima Boudaoud *"A Semantic Engine for Internet of Things: Cloud, Mobile Devices and Gateways"* 4th International Workshop on Extending Seamlessly to Internet of Things, in conjunction with the 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2015), 8-10 July 2015, Blumenau, Santa Carina, Brazil
- Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet and Karima Boudaoud *"Standardizing Generic Cross-Domain Applications in Internet of Things"* Third Workshop on Telecommunications Standards 2014, Part of IEEE Globecom, 8-12 December 2014, Austin, TX, USA

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud. *"Helping IoT application developers with sensor-based linked open rules"* In SSN 2014, 7th International Workshop on Semantic Sensor Networks in conjunction with the 13th International Semantic Web Conference (ISWC 2014), 19-23 October 2014, Riva Del Garda, Italy

A.3 Doctoral Consortiums

- Amelie Gyrard. *"A machine-to-machine architecture to merge semantic sensor measurements"* In Proceedings of the 22nd international conference on World Wide Web companion, pages 371-376. International World Wide Web Conferences, May 13-17 2013, Rio de Janeiro, Brazil

- Amelie Gyrard, Christian Bonnet and Karima Boudaoud *"An architecture to aggregate heterogeneous and semantic sensed data"* 10th Extended Semantic Web Conference, PhD Symposium, (ESWC 2013), 26-30 May 2013, Montpellier, France

A.4 Posters

- Amelie Gyrard, Christian Bonnet and Karima Boudaoud *"Ontology-based Intelligent Transportation Systems"* BMW Summer School 2014, Autonomous Driving in the Internet of Cars, July 27 - August 1 2014

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud. *The stac (security toolbox: attacks & countermeasures) ontology"* In Proceedings of the 22nd international conference on World Wide Web companion, pages 165-166. International World Wide Web Conferences, May 13-17 2013, Rio de Janeiro, Brazil

- Amelie Gyrard, Christian Bonnet and Karima Boudaoud *"STAC: Un outil pour vous aider à sécuriser vos applications"* SAR-SSI 2013, 8ème Conférence sur la Sécurité des Architectures Réseaux et des Systèmes d'Information, 16-19 September 2013, Mont De Marsan, France

A.5 Participation to Standards

- Amelie Gyrard, Christian Bonnet *"A unified language to describe M2M/IoT data"* OneM2M, MAS Working Group 5, 22-27 March 2015, Sophia Antipolis, France

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud *"Domain knowledge Interoperability to build the Semantic Web of Things"* W3C Workshop on the Web of Things Enablers and services for an open Web of Devices, 25-26 June 2014, Berlin, Germany

- Amelie Gyrard, Christian Bonnet *"Semantic Web best practices: Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things"* OneM2M International standard, Working Group 5 (Management, Abstraction and Semantics), April 2014

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud *"An Ontology to Semantically Annotate the Machine-to-Machine (M2M) Device Measurements"* ETSI M2M Workshop 201, 5-7 November 3013, Mandelieu, France

A.6 Under Reviews

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud *"LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications"* IEEE Internet of Things Journal 2015 (submitted in May 2014, revised manuscript resubmitted in November 2014, second review process)

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud *"Semantic Web of Things: A Survey"* IEEE Communications Surveys & Tutorials 2015 (submitted in February 2015, under review)

- Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, Karima Boudaoud *"Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation"* 3rd International Conference on Future Internet of Things and Cloud (FiCloud 2015), 24-26 August 2015, Rome, Italy

- Amelie Gyrard, Christian Bonnet, Karima Boudaoud *"SWoT Generator: Assisting IoT Projects in Designing Interoperable Semantic Web of Things Applications"* 5th International Conference on the Internet of Things (IoT 2015), 26-28 October 2015, Seoul, Korea

- Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, Karima Boudaoud *"Integrating Machine-to-Machine Measurement Framework into oneM2M Architecture"* 17th Asian-Pacific Network Operations and Management Symposium (APNOMS 2015), 19-21 August 2015, Busan, Korea

- Amelie Gyrard, Christian Bonnet, and Karima Boudaoud *"Smarter LOV4IoT dataset: Exploiting, Reusing and Combining Domain Knowledge Expertise"* 14th International Semantic Web Conference (ISWC 2015), Data Sets and Ontologies track, October 11-15 2015, Bethlehem, Pennsylvania, USA

- Soumya Kanti Datta, Amelie Gyrard, Christian Bonnet, Karima Boudaoud *"oneM2M Architecture based User Centric IoT Application Development"* 3rd International Conference on Future Internet of Things and Cloud (FiCloud 2015), 24-26 August 2015, Rome, Italy

Appendix B

Abbreviations & Glossary

B.1 Abbreviations

IoT	Internet of Things
WoT	Web of Things
SWoT	Semantic Web of Things
M2M	Machine-to-Machine
SWE	Sensor Web Enablement
SenML	SensorML
RDF	Resource Description Framework
RDFS	RDF Schema
OWL	Ontology Web Language
SSN	Semantic Sensor Network
XML	Extensible Markup Language
JSON	JavaScript Object Notation
LOV	Linked Open Vocabularies
LOV4IoT	Linked Open Vocabularies for Internet of Things
LOD	Linked Open Data
LOR	Linked Open Rules

S-LOR	Sensor-based Linked Open Rules
M3	Machine-to-Machine Measurement
RIF	Rule Interchange Format
SWRL	Semantic Web Rule Language
API	Application Programming Interface
XSLT	Extensible Stylesheet Language Transformations
JAXB	Java Architecture for XML Binding
ECC	Elliptic Curve Cryptography
MAC	Message Authentication Code
AES	Advanced Encryption Standard
ECG	Electrocardiogram
Wi-Fi	Wireless-Fidelity
W3C	World Wide Web Consortium
SSL	Secure Socket Layer
XSLT	Extensible Stylesheet Language Transformations
VPN	Virtual Private Network

DUL	Dolce Ultra Lite
IDS	Intrusion Detection Systems
OSI	Open Systems Interconnection model
UMTS	Universal Mobile Telecommunication System
GSM	Global System for Mobile Communication
GPRS	General Packet Radio Service
WSN	Wireless Sensor Networks

B.2 Glossary

M3 template	A package provided to the developer with ontologies, datasets, rules and SPARQL queries required to build a Semantic Web of Things applications
Jena	A framework to build semantic web applications
SPARQL	SQL-like language
Triple store	Database to store semantic data called triplet
SPARQL endpoint	Semantic data can be directly accessible online via SPARQL queries
SenML	A language to describe sensor measurements adapted to constrained devices
Ontology	Design in a structure way a domain by describing concepts and their relationships
Concept	Concepts are designed in ontologies (OWL) Similar to the idea of classes on Object-Oriented Proagramming
Instance	Instances (also called individuals) are designed in datasets (RDF). Instances are related to concepts
X_509	Used to deliver certificated required to exchange keys and encrypt data
Security by design	The software has been designed from the ground up to be secured

Appendix C

French Summary 20 pages

Conception des applications Internet des Objets sémantiques inter-domaine

Amélie Gyrard

A doctoral dissertation submitted to:
TELECOM ParisTech
In Partial Fulfillment of the Requirements for the Degree of:
Doctor of Philosophy
Specialty : Computer Science

Thesis Supervisor: **Prof. Christian Bonnet**
Thesis Co-Supervisor: **Dr. Karima Boudaoud**

Jury:
Président:
Prof. Bruno Martin - Université Nice Sophia-Antipolis, Sophia Antipolis - France

Rapporteurs:
Dr. Jérôme Euzenat - INRIA, Grenoble - France
Prof. Oscar Corcho - Universidad Politecnica de Madrid, Madrid - Spain

Examinateurs:
Dr. Payam Barnaghi - University of Surrey, Guildford - United Kingdom
Claude Hary - Com4Innov, Sophia Antipolis - France

Invité:
Philippe Badia - Com4Innov, Sophia Antipolis – France

Résumé

Selon les prévisions de Cisco[1], il y aura plus de 50 milliards d'appareils connectés à Internet d'ici 2020. Les appareils et les données produites sont principalement exploitées pour construire des applications « Internet des Objets (IdO) ». D'un point de vue des données, ces applications ne sont pas interopérables les unes avec les autres. Pour aider les utilisateurs ou même les machines à construire des applications 'Internet des Objets' inter-domaines innovantes, les principaux défis sont l'exploitation, la réutilisation, l'interprétation et la combinaison de ces données produites par les capteurs. Pour surmonter les problèmes d'interopérabilité, nous avons conçu le système Machine-to-Machine Measurement (M3) consistant à: (1) enrichir les données de capteurs avec les technologies du web sémantique pour décrire explicitement leur sens selon le contexte, (2) interpréter les données des capteurs pour en déduire des connaissances supplémentaires en réutilisant autant que possible la connaissance du domaine définie par des experts, et (3) une base de connaissances de sécurité pour assurer la sécurité dès la conception lors de la construction des applications IdO. Concernant la partie raisonnement, inspiré par le « Web de données », nous proposons une idée novatrice appelée le « Web des règles » afin de partager et réutiliser facilement les règles pour interpréter et raisonner sur les données de capteurs. Le système M3 a été suggéré à des normalisations et groupes de travail tels que l'ETSI M2M, oneM2M, W3C SSN et W3C Web of Things. Une preuve de concept de M3 a été implémentée et est disponible sur le web (http://www.sensormeasurement.appspot.com/) mais aussi embarqué dans des appareils mobiles tels que les téléphones ou les tablettes.

I. Introduction

1. Motivation

Au cours des dernières années, nous avons assistés à un nombre croissant de capteurs intégrés à de nombreux appareils tels que les téléphones mobiles, les montres ou des lunettes intelligentes ainsi que des puces intégrées à des objets de tous les jours tels que les livres, les vêtements, CDs, DVDs, etc... Les applications exploitant ces capteurs ainsi que les données produites sont de plus en plus populaires. La domotique ainsi que les applications liées au bien-être sont de plus en plus présentes dans notre vie quotidienne. Par exemple, Hapifork[2] suit votre habitudes alimentaires. Oral-B[3] et Kolibree[4] sont des brosse à dents intelligentes connectées à Internet afin de vérifier l'hygiène dentaire. Mother[5] vérifie si vous marchez assez pour rester en forme, rappelle les médicaments, surveille la qualité de votre sommeil, etc. L'Apple HealthKit[6] surveille votre santé, la nutrition et le sommeil, etc. SFR Connected Homes[7] propose des thermostats et éclairages connectés afin de les contrôler à distance. Les capteurs sont également déployés dans des fermes et jardins intelligents. Edyn[8] et Botanicalls[9] envoient des alertes lorsque les plantes ont besoin

[1] http://share.cisco.com/internet-of-things.html
[2] http://www.hapi.com/product/hapifork
[3] http://connectedtoothbrush.com/
[4] http://www.kolibree.com/
[5] https://sen.se/store/mother/
[6] http://goo.gl/n2V42g
[7] http://connected-objects.fr/2014/05/sfr-home-box-domotique/
[8] https://www.kickstarter.com/projects/edyn/edyn-welcome-to-the-connected-garden

d'être arrosées. Les Google Cars[10], des voitures sans conducteurs sont déjà autorisées en ville dans le Nevada ou en Californie. Les appareils intelligents sont de plus en plus connectés à Internet et leur données sont envoyées sur le Web pour construire l'« Internet des objets (IdO) », plus précisément le « Web des choses ». Selon les prédictions[11] de Cisco, il y aura plus de 50 milliards d'appareils connectés à Internet d'ici 2020. En raison de l'énorme quantité de données de capteur produit, il y a un réel besoin d'interpréter ces données et de construire des applications de l'IdO interopérables.

2. **Problème**

Un des problèmes difficiles est dû aux appareils qui ne sont pas interopérables les uns avec les autres : leur données sont basées sur des formats propriétaires et ils n'utilisent pas de termes ou de vocabulaires communs pour décrire des données interopérables. Le problème est similaire avec les applications car elles sont basées sur des protocoles propriétaires. Une façon de rendre ces applications interopérables serait un protocole commun utilisé par tous les appareils. Une autre solution serait de travailler sur l'interopérabilité de ces données, puisque ces appareils sont déjà déployés et les données sont déjà produites. Exploiter, combiner et enrichir les données produites par les capteurs permettraient de créer des applications plus intelligentes, et surtout interopérables les unes avec les autres, ce qui est un véritable défi. Le "Web de données" est de plus en plus employé et encourage les gouvernements et autres organisations à partager les données sur le Web, y compris les données issues des capteurs. Pour aider les utilisateurs et même les machines à l'interprétation et à la fusion de ces données, il y a un réel besoin de décrire explicitement les mesures de capteurs en fonction du contexte, de façon unifiée et étant compréhensible par des machines. Par exemple, une mesure de température n'a pas le même sens selon le contexte (température ambiante, température corporelle, température de l'eau ou température externe) et la machine ne déduira pas les mêmes connaissances (la fièvre est déduite avec la température du corps, une température anormale pour une température ambiante). Nous devons également faire face à des unités implicites (i.e., Fahrenheit, Celsius, Kelvin).

Le deuxième plus grand défi est de combiner des applications et des données issues de domaines hétérogènes ensemble afin de créer des applications IdO innovantes et inter-domaines. Les applications existantes sont spécifiques pour un domaine particulier tels que la maison intelligente, les soins de santé à distance, le transport, les agricultures intelligentes, etc. La Figure 1 montre des exemples d'applications innovantes combinant les domaines: (1) suggestions alimentaires selon la météo, (2) suggestions de remèdes maison quand vous avez de la fièvre, et (3) suggestions d'équipements dans une voiture intelligente selon le météo, etc. Un frigidaire intelligent permettrait d'acheter des ingrédients en ligne. En cas de puces RFID intégrées aux aliments, il sera facile de recommander le menu pour le dîner ou commander automatiquement des ingrédients manquants. Si vous êtes un athlète, le frigidaire intelligent vous conseillera le régime parfait [9], en prenant compte des données envoyées par les chaussures intelligentes (ex., chaussures Nike compatible avec le système Apple[12]). Enfin, la voiture de Google pourrait vous conduire automatiquement à l'épicerie pour acheter les ingrédients manquants. Un des défis les plus importants pour l'internet des objets serait d'aider les

[9] http://www.botanicalls.com/
[10] http://en.wikipedia.org/wiki/Google_driverless_carandBMWconnectedcars
[11] http://share.cisco.com/internet-of-things.html
[12] https://www.apple.com/fr/ipod/nike/

développeurs dans la conception et le développement d'applications IdO interopérables qui fusionnerait également les domaines les uns avec les autres.

Enfin, les questions de sécurité doivent être considérées lors de la fusion des données ou lors de la conception d'applications pour l'Internet des Objets. Par exemple, les données de santé sont plus sensibles comparées aux données issues de la météo.

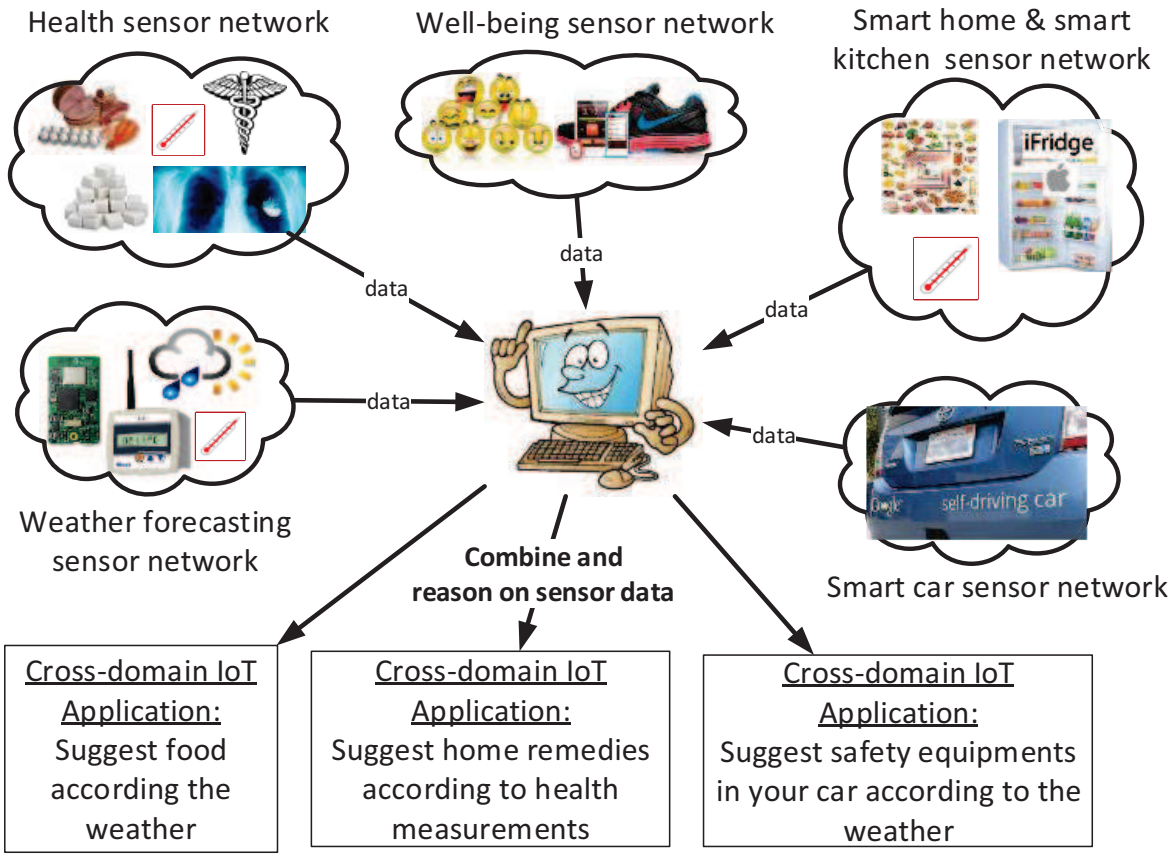

Figure 1. Combiner les domaines afin de concevoir des applications plus intelligentes

3. Notre approche

Dans cette thèse, nous nous concentrons sur l'interopérabilité des données de capteurs afin de construire des applications interopérables qui de plus permettraient de fusionner des domaines hétérogènes. Pour faire face à ce défi, nous exploitons les technologies du web sémantique pour plusieurs raisons [3]. Tout d'abord, la sémantique permet une description explicite de la signification des données de capteurs. Ainsi, les machines peuvent comprendre, fusionner et interpréter les données. Deuxièmement, les technologies du web sémantique facilitent l'interopérabilité des données et leur intégration puisque les données issues de différents capteurs seront converties selon le même vocabulaire.

Troisièmement, les moteurs de raisonnement sémantiques peuvent être facilement utilisés pour en déduire des abstractions de haut niveau à partir de données de capteurs. Quatrièmement, la sensibilité au contexte pourrait être implémentée en utilisant le raisonnement sémantique. Enfin, en théorie, la sémantique facilite le partage des connaissances et la réutilisation de l'expertise du

domaine ce qui permettrait d'éviter de constamment recoder la même connaissance du domaine lors du développement d'une nouvelle application. En effet, chaque fois qu'une nouvelle application est créée, un nouveau vocabulaire est défini. Les technologies du web sémantique deviennent très populaires et sont adoptées par les entreprises tels que Google et Yahoo. De nombreux sites Web utilisent Schema.org[13] pour améliorer les résultats de la recherche dans les moteurs de recherches. Schema.org est un ensemble de vocabulaires appelés ontologies pour décrire les données sur le Web d'une manière unifiée. Par exemple, ils ont définis des vocabulaires communs pour décrire les personnes, les organisations, etc… Dans la Figure 1Figure 2, à droite, le moteur de recherche reconnaît automatiquement que Steve Jobs est une personne, sa conjointe est Lauren Powell Jobs, etc. Google introduit l'idée des graphes de connaissances afin de structurer et de connecter les données les unes avec les autres. Nous proposons une approche similaire afin de structurer et combiner les données des capteurs afin de construire des d'applications Internet des Objets inter-domaines.

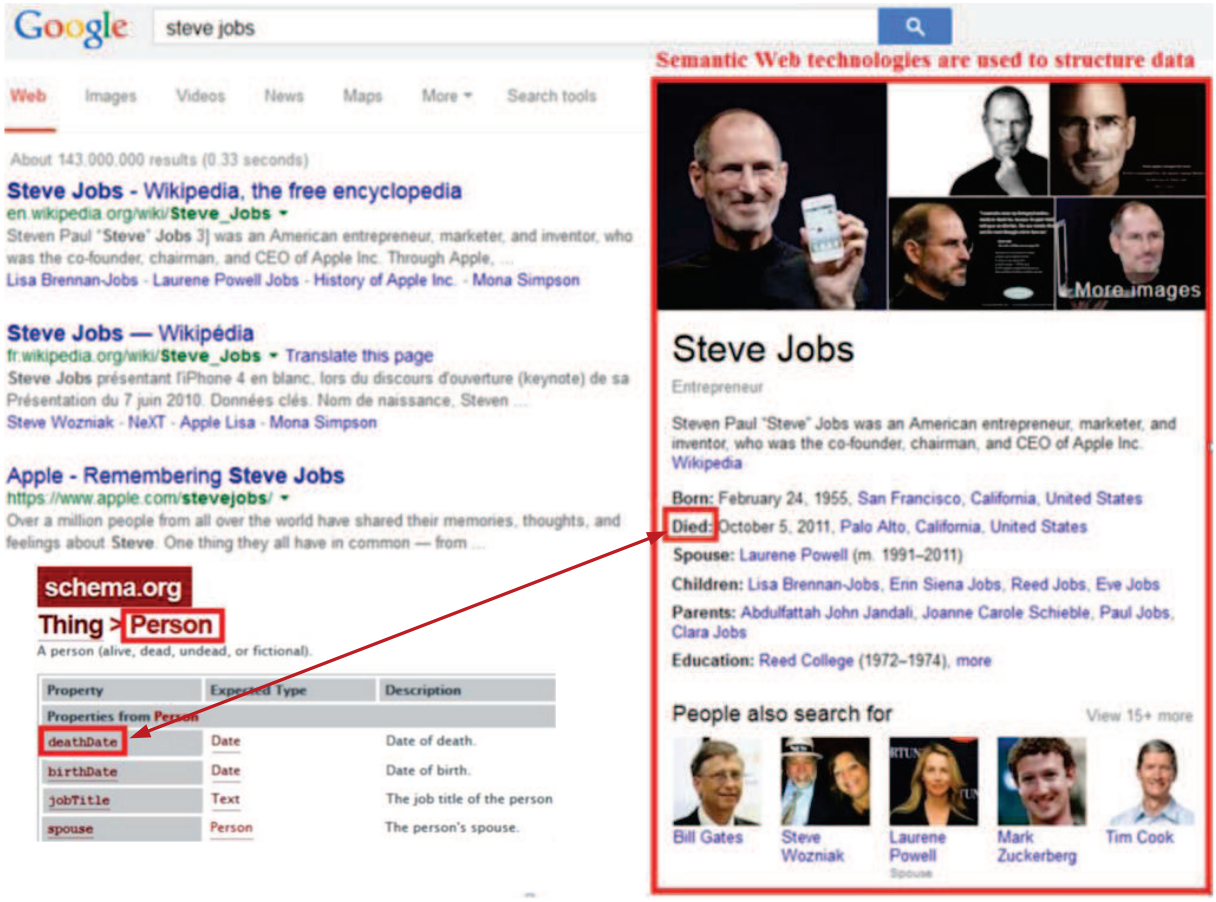

Figure 2. Les technologies du Web sémantique utilisée par Google pour structure les données sur le Web

En outre, selon P. Barnaghi et ses co-auteurs, la sémantique est nécessaire dans différentes couches de l'IdO, elle peut être utilisée pour: (1) décrire les données, (2) réutiliser la connaissance du domaine, (3) interpréter les données de l'IdO, (4) concevoir des applications plus intelligentes, et (5) assurer la sécurité [2]. Dans cette thèse, nous abordons les défis suivants:

[13] http://schema.org/

- La génération d'applications IdO inter-domaines interopérables fondée sur la sémantique. Ce processus devrait être assez souple pour être réalisé soit sur le 'cloud', ou intégré dans des dispositifs restreints tels que les téléphones mobiles.
- L'interprétation des données de capteurs afin de déduire de nouvelles connaissances en réutilisant l'expertise du domaine. L'interopérabilité des connaissances de domaine permet de construire une expertise inter-domaine.
- La sécurisation des applications IdO lors de la conception de ces applications. Ce défi devra être résolu en utilisant la même approche que pour les deux défis précédents en combinant l'expertise définie par les experts en sécurité afin de sélectionner les mécanismes de sécurité les plus appropriés pour sécuriser les applications.

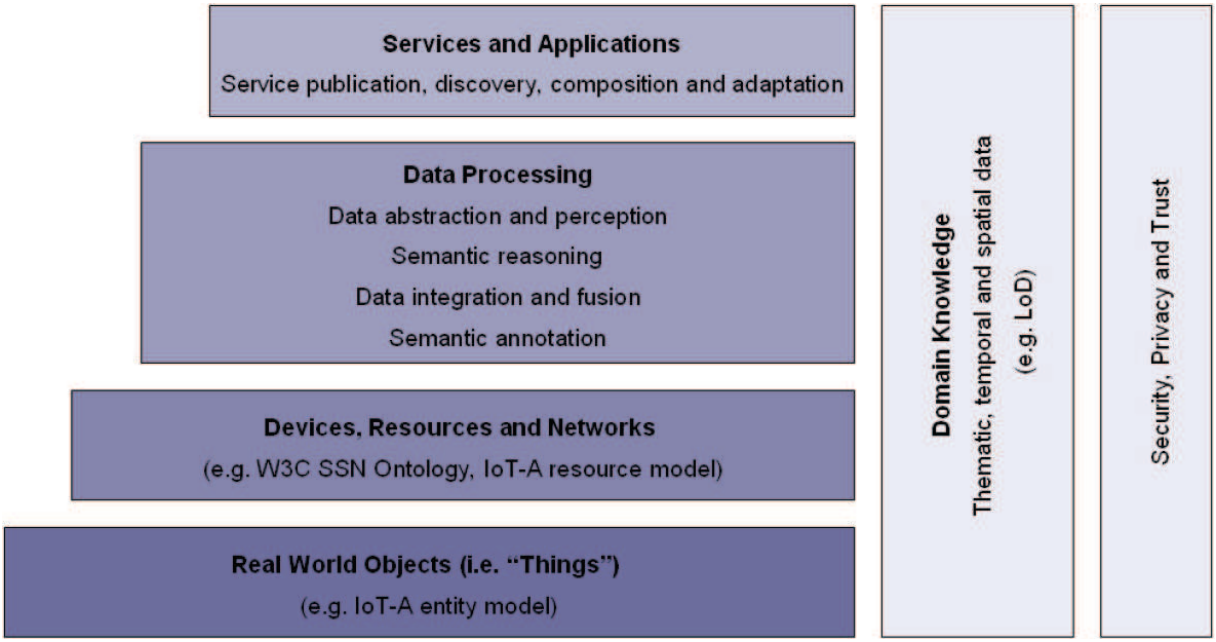

Figure 3. Sémantique requise dans l'Internet des Objets [2]

4. Hypothèse

Dans cette thèse, nous supposons plusieurs hypothèses. Tout d'abord, nous supposons qu'un prétraitement est fait pour nettoyer les données issues des capteurs (par exemple, les données peu fiables au cas où un capteur est mort) et un second prétraitement qui ajoute des métadonnées aux données produites par les capteurs tels que le type de mesure, l'unité, le type de capteur, la valeur et le domaine. Par exemple, les données de capteur sont représentées de cette façon: si le domaine est la santé, le type de mesure peut être la température, le valeur peut être 39 et l'unité degré Celsius. D'autre part, nous considérons que les données de capteur sont générées par de simples capteurs tels que le thermomètre et non par des capteurs compliqués tels que l'électrocardiogramme (ECG). Enfin, dans ce travail, nous ne considérons pas les défis suivants: le temps réel, le passage à l'échelle et l'analyse de grandes quantités de données.

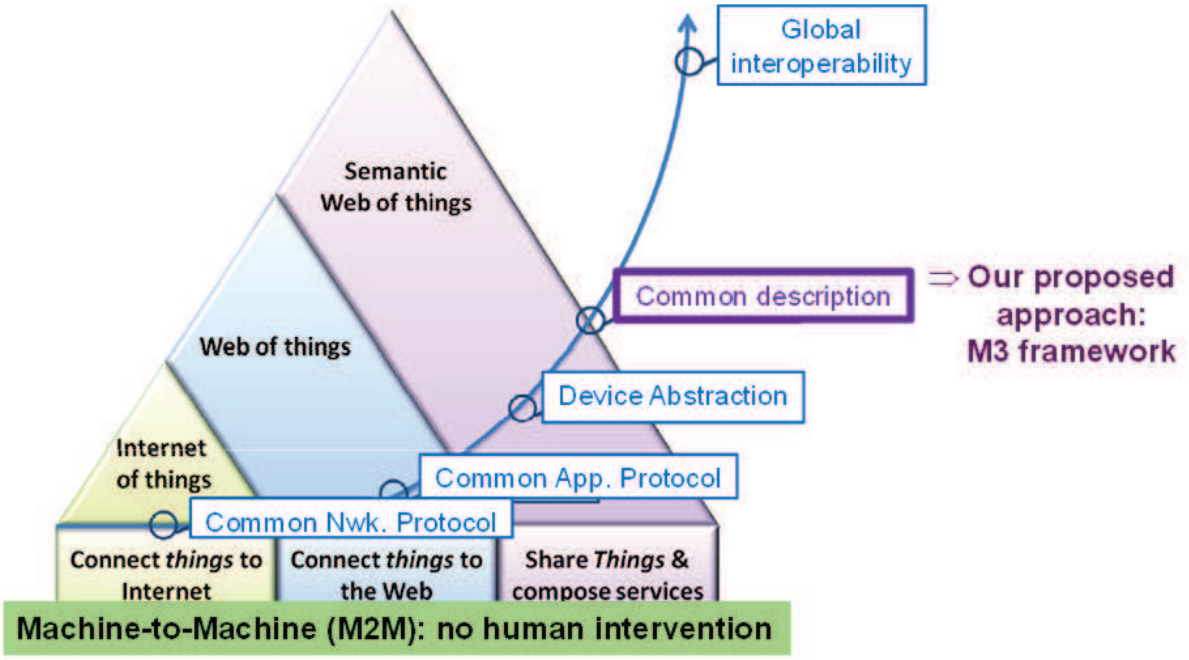

Figure 4. Prochains challenges dans l'Internet des Objets sémantique [6]

5. Contributions

Pour aider les développeurs dans la conception et la mise en œuvre d'applications internet des Objets inter-domaines basées sur les technologies du Web Sémantique, nous avons conçu le système **Machine-to-Machine Measurement (M3)**. Récemment, A. Jara et ses co-auteurs ont expliqué que les prochains enjeux concernant le Web sémantique des choses (voir Figure 1.4) sont: (1) une description commune pour les données de capteurs, et (2) être d'accord sur un catalogue commun d'ontologies pour annoter les données des capteurs de manière interopérable [6]. Ces défis complexes ont été résolus grâce à notre système M3.

La contribution principale de cette thèse est le système M3 qui permet d'interpréter automatiquement les données issues des capteurs et de mettre à disposition une expertise du domaine interopérable afin de concevoir des applications IdO inter-domaines. Dans cette thèse, nous expliquons en détail les divers composants du système M3 et la façon dont ils sont interconnectés les uns avec les autres. Par ailleurs, l'approche proposée peut être intégrée dans une plate-forme Machine-to-Machine (M2M), appelée Com4Innov, déployée à Sophia Antipolis en France. Une perspective était d'étendre cette plateforme avec de nouvelles fonctionnalités telles que l'ajout d'une intelligence aux données avec la sémantique. Pour cette raison, nous intégrons le système M3 dans des normalisations telles que l'ETSI M2M [7]. En effet, nous proposons une extension à l'architecture M2M proposée par la normalisation ETSI M2M en y ajoutant de la sémantique. Nous avons développés plusieurs cas d'applications inter-domaines tels que la naturopathie, le tourisme, le transport et la sécurité. Ces application ont été mis en œuvre comme preuve de concepts sur le 'cloud' mais aussi embarquées sur des dispositifs restreints comme les téléphones mobiles ou les tablettes.

Notre deuxième contribution est une idée innovante qui permet le partage, la réutilisation et la combinaison de règles interopérables afin d'interpréter les données de capteurs. Nous avons nommées ce composant **Sensor-based Linked Open Rules (S-LOR)**. S-LOR est une extension

au ‘Web de données’ qui permet le partage et la réutilisation des données. Grâce à S-LOR, nous pouvons également partager et réutiliser les règles pour interpréter les données issues de capteurs. Nous annotons sémantiquement les données des capteurs (par exemple, au format SenML) avec la sémantique. Pour cela, nous avons conçu, la nomenclature M3 implémentée dans une ontologie afin d'agréger et de décrire de manière uniforme les capteurs, les mesures, les unités et les domaines. Nous proposons également d’interpréter les données de capteurs et de déduire des abstractions de haut niveau en exploitant l'expertise des experts du domaine. Cette expertise est extraite d’un catalogue appelé **Linked Open Vocabularies for Internet of Things (LOV4IoT)**. Nous avons conçu le catalogue LOV4IoT afin de référencer, synthétiser, classer et réutiliser plus de 200 projets dans divers domaine pertinents pour l’Internet des Objets tels que la santé, la domotique, le tourisme, les voitures intelligentes, les agricultures intelligentes, etc... Ces projets utilisent également les technologies du web sémantique et nous souhaitons récupérer leur ontologies et leur règles. Dû à de nombreux problèmes d'interopérabilité pour combiner les connaissances du domaine, nous avons dû réécrire la connaissance de manière interopérable : les ontologies, les règles, et les bases de connaissances. Cette tâche était essentielle afin de montrer l'ensemble de la chaîne du traitement M3. Cette nouvelle base de connaissance interopérable facilite la raisonnement sur les données des capteurs ainsi que l'interconnexion inter-domaines (par exemple, météo et le transport, la météo et la maison intelligente) pour générer des applications IdO innovantes inter-domaines.

Notre troisième contribution principale est **Security Toolbox : Attacks & Countermeasures (STAC)** qui aide les utilisateurs à choisir des mécanismes de sécurité les plus pertinents pour sécuriser leur applications. L'approche STAC réutilise plusieurs composants fournis par le système M3. Par exemple, nous avons conçu STAC en utilisant la même approche que pour la conception de la base de connaissance pour interpréter les données des capteurs. STAC est une base de connaissance de sécurité qui classifie les attaques et les mécanismes de sécurité dans divers domaines tels que les réseaux de capteurs, les réseaux sans fil, l’administration réseau, les applications Web, etc…

Les trois contributions principales sont validées par cinq cas d'utilisation. L'objectif du premier cas d'utilisation est de montrer que le système M3 peut être utilisé par les développeurs et est souple assez pour être intégré sur des appareils Android. Dans le second cas d'utilisation, nous montrons comment M3 peut être utilisé dans des voitures intelligentes. Dans le troisième cas d'utilisation, nous démontrons comment M3 peut être intégré dans les réfrigérateurs intelligents. Dans le quatrième cas d'utilisation, nous pouvons expliquer comment M3 peut être intégré dans des bagages intelligents. Enfin, dans le cinquième cas d'utilisation nous expliquons comment STAC peut aider les développeurs non-expert en sécurité à choisir le mécanisme de sécurité nécessaire pour sécuriser leur application.

II. State of The Art

Nous avons étudions le Web sémantique des objets (SWoT) et les différent domaines liés à ce sujet de recherche. Nous avons analysés et synthétisés de nombreux domaines tels que Ubiquitous Computing (UbiComp) ou informatique ubiquitaire, Pervasive Computing, intelligence ambiante (AMI), Context-Awareness, Ambient Assisted Living (AAL), les maisons intelligentes, les réseaux de capteurs sémantiques (SSN), Machine-to-Machine (M2M), Internet des objets (IdO), Web of Things (WoT), Web sémantique des objets (SWOT), les villes intelligentes et le Physical-Cyber-Social Computing (PCS). Ensuite, nous avons mis en évidence

des principaux défis à relever et nous avons examiné les travaux et outils dans chacun de ces défis : (1) les données de capteurs interopérables, (2) l'interprétation des données, (3) l'interopérabilité inter-domaine, (4) aider les développeurs à concevoir des applications IdO basée sur la sémantique, (5) les mécanismes 'Sensor Plug & Play afin de reconnaitre automatiquement les nouveaux capteurs connectés, (6) la sémantique appliquées aux dispositifs restreints, et (7) les problèmes de sécurité pour l'IdO. Enfin, nous avons résumés les limites actuelles de l'état de l'art et élucidons des solutions.

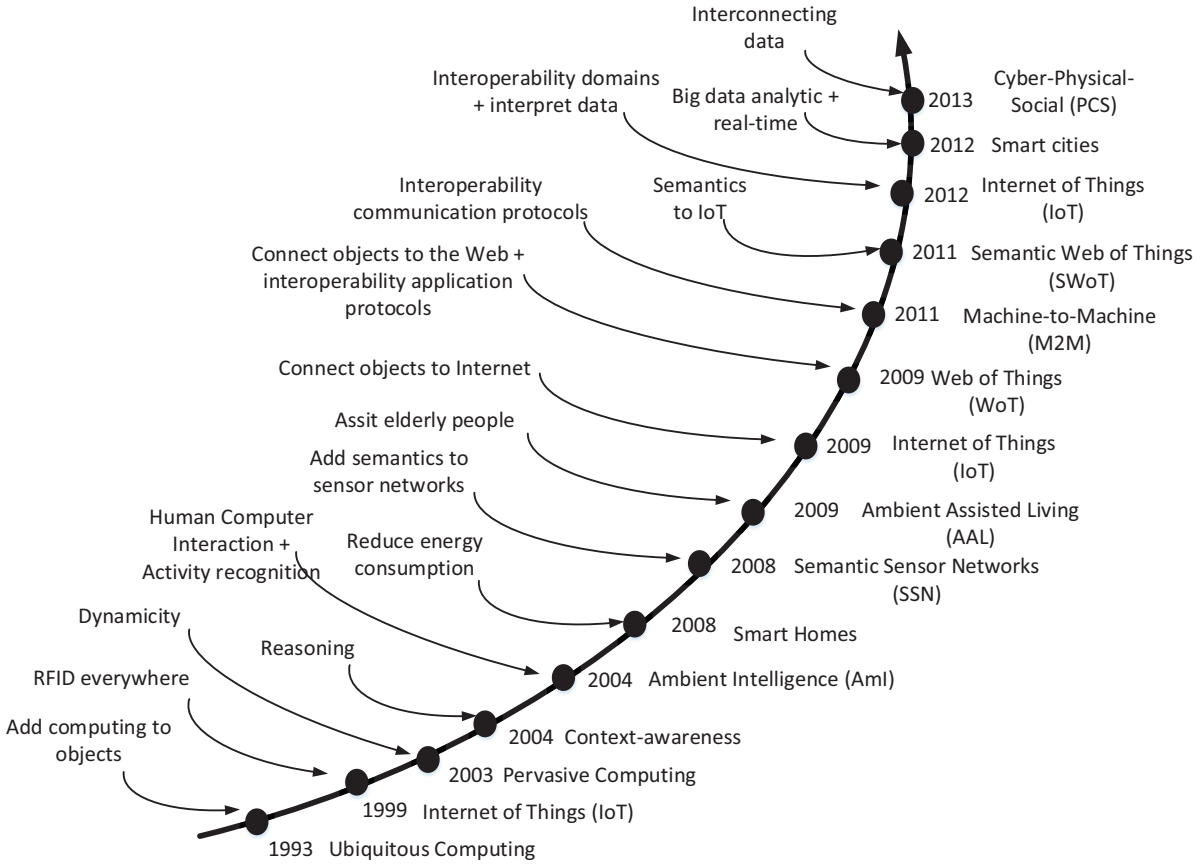

Figure 5. Évolution des domaines de recherché liés au Web sémantique des Choses

Basée sur l'analyse des limitations des travaux existants les principaux défis doivent être relevés :

- Défi A : L'interopérabilité des données IdO sera adressé grâce au langage et à l'ontologie M3.
- Défi B : Interprétation des données IdO sera adressé avec S-LOR en réutilisant les connaissances du domaine référencées dans LOV4IoT.
- Défi C : L'interopérabilité inter-domaine sera adressé avec les connaissances du domaine interopérable M3 et LOV4IoT.

- Défi D : Conception d'applications interopérables SWoT sera adressé avec le système M3 et le générateur SWoT.
- Défi E : Sécurisation des applications IdO sera adressé avec STAC.

III. M3 Framework

Dans cette section, nous aidons les développeurs IdO à exécuter facilement cette tâche "Défi D: Conception d'applications Internet des Objets Sémantiques (IdOS) interopérables". Nous supposons dans ce travail que les développeurs veulent concevoir des applications de l'IdO afin d'interpréter les données de capteurs. Fréquemment, les développeurs IdO réalisent quatre tâches comme illustré dans la figure: (1) concevoir des applications SWoT, (2) annoter sémantiquement données IdO, (3) interpréter les données de l'IdO, et (4) sécuriser les applications IdO. Ce framework permet aux développeurs d'utiliser les technologies et outils du web sémantique sans avoir à se familiariser avec, ce qui leur facilite beaucoup la tâche et leur fait économiser du temps. De plus, plus les développeurs utiliseront, plus les données et les suggestions produits seront interopérable. Dans toutes ces étapes, notre objectif est de simplifier autant que possible le travail des développeurs tout en fournissant l'unification.

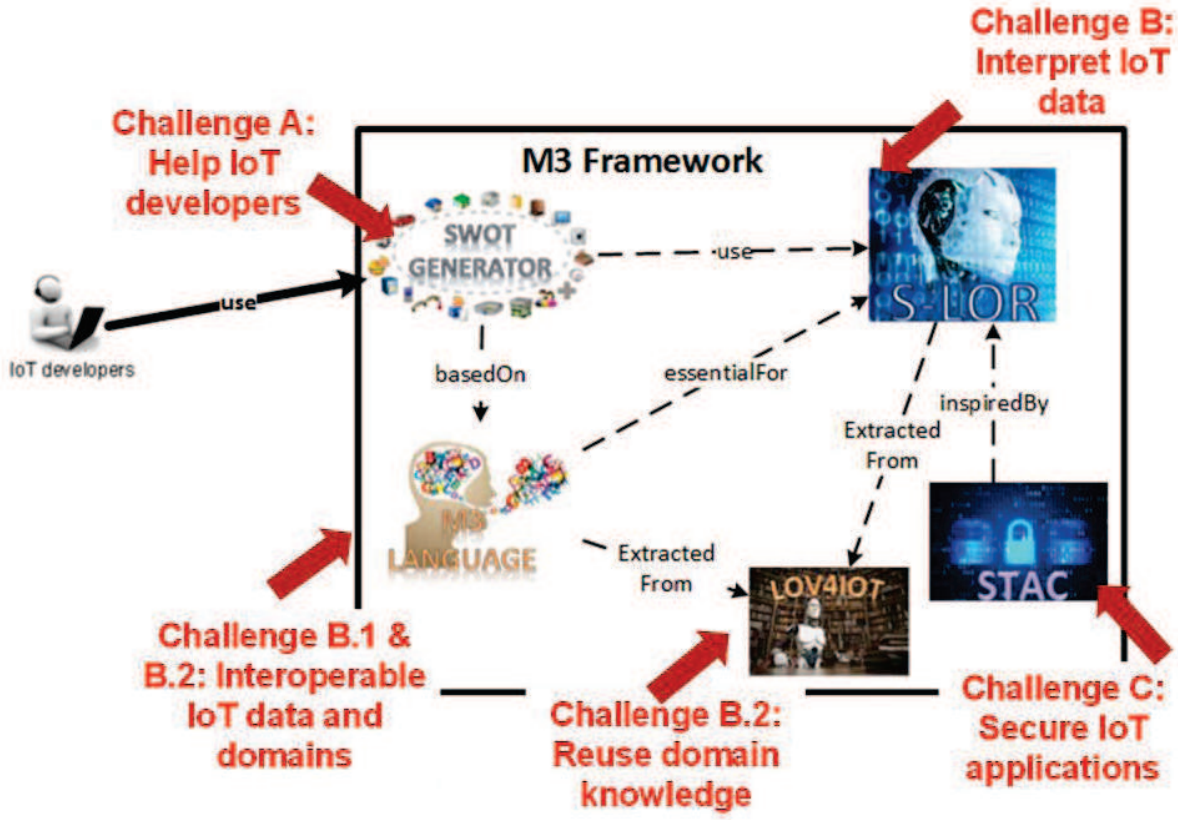

Figure 6. Le système M3 et sous-composants adressant les differents défis

1. SWoT generator

Le système M3 est composé du générateur SWoT pour produire des modèles qui seront utilisés pour concevoir facilement des applications SWoT (voir Figure 7). SWoT est un générateur, car il va générer des modèles M3 aux développeurs avec des ontologies, des bases de connaissances de domaines, des règles et les requêtes SPARQL nécessaires pour construire leurs applications IdO. L'importance capitale est que tous ces éléments sont interopérables les uns avec les autres. Le principal bénéfice du modèle M3 est d'éviter aux développeurs d'apprendre les technologies du web sémantique, plus précisément, ils n'ont pas besoin de concevoir leurs propres ontologies et règles ou même d'annoter sémantiquement les données. Les modèles M3 ont été conçus manuellement et sont regroupés dans une base de données pour construire des applications IdO populaires. Les développeurs indiquent le nom du capteur utilisé (par exemple, Luminosité) et le domaine (par exemple, Météo) et le générateur SWoT cherche les modèles M3 en fonction de ces critères en exécutant une requête SPARQL sur la base contenant les modèles M3. Le générateur produit le modèle M3 composé des ontologies de domaine, des bases de connaissances, des règles et des requêtes SPARQL. Ces modèles permettront facilement d'annoter sémantiquement les données IdO, les interpréter et fournir des suggestions.

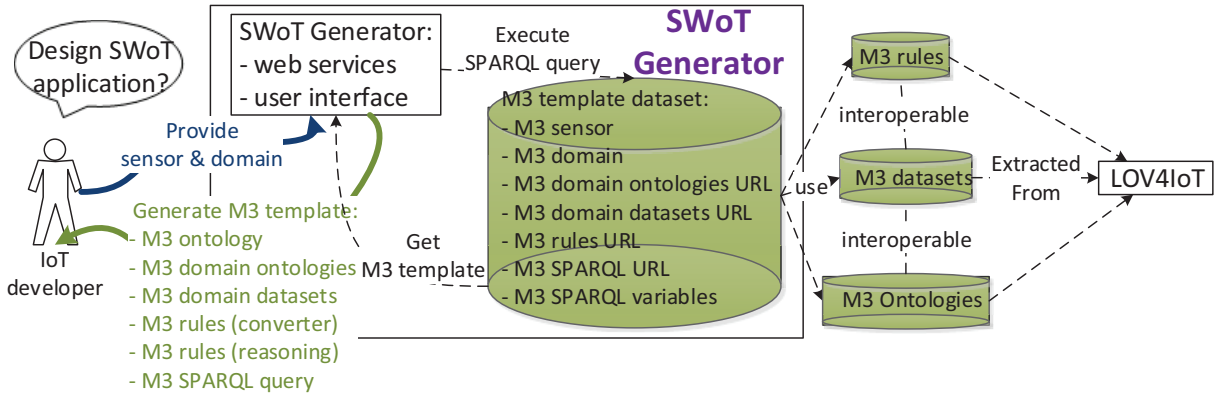

Figure 7. Obtenir des modèles M3 avec le générateur SWoT

2. M3 langage

Le langage M3 est essentiel pour fusionner et combiner les données provenant de projets ou domaines hétérogènes. De plus, c'est une étape essentielle pour le raisonnement. Pour cette raison, nous avons conçu le langage M3 afin de rendre les données interopérables et d'ajouter explicitement le contexte si nécessaire pour supprimer toute ambiguïté. Par exemple, une température peut être la température du corps ou une température extérieure. Pour atteindre cet objectif, nous avons utilisé, plus précisément, nous avons conçu l'ontologie M3 pour les raisons suivantes:

- Faciliter l'interopérabilité.
- Ajouter des descriptions explicites de métadonnées du capteur.
- Employer un moteur de raisonnement afin de déduire de nouvelle connaissances.
- Réutiliser la connaissance du domaine.
- Fournir une manière souple et facile pour mettre à jour la nomenclature M3.

La nomenclature M3 a été mis en œuvre dans une ontologie afin de fournir une base pour le raisonnement et l'interconnexion des domaines. L'ontologie M3 synthétise et unifie les termes pour décrire les capteurs, les mesures, les actionneurs et les domaines trouvés dans les projets

existants référencé dans LOV4IoT. Par exemple, les précipitations et les capteurs de pluie représentent le même capteur. Les descriptions uniformes mentionnés ci-dessus sont fondamentalement nécessaire afin de développer des applications interoperables et des services inter-domaines. Une nomenclature commune est décrite ci-dessous et la liste n'est pas exhaustive. Le langage M3 est une extension de l'ontologie W3C SSN. Plus précisément, une extension des concepts ObservationValue, FeatureOfInterest et Sensor. En outre, le langage M3 permet de décrire les mesures de capteurs d'une manière interopérable. Une mesure a un nom (par exemple, la température), une valeur (Par exemple, 39) et une unité (par exemple, DegreeCelcius). Nous ne traitons pas uniquement les capteurs mais également les puces RFID, les actionneurs, etc. Les descriptions uniformes du capteur, des mesures et les domaines ont déjà été communiquées au standardisations telles que oneM2M WG5 MAS [4].

3. S-LOR

Dans la Figure 8, le développeur utilise le convertisseur M3 afin d'annoter sémantiquement les données fournies par les capteurs. Les données converties sont alors compatible avec la nomenclature et l'ontologie M3, une étape essentielle pour faciliter l'interprétation des données. Ensuite, les données au format M3 sont enrichies grâce à la connaissance du domaine interopérable que nous avons réécrite qui est composée d'ontologies de domaine, des bases de données et des règles compatibles les uns avec les autres. Ainsi, le processus de raisonnement permet d'enrichir les données pour déduire des abstractions de haut niveau, de proposer des suggestions et de fusionner les domaines aux développeurs. Enfin, le développeur affichera les résultats dans une interface conviviale, pour envoyer des alertes ou actionner certain actionneurs (par exemple, ouvrir ou fermer une porte).

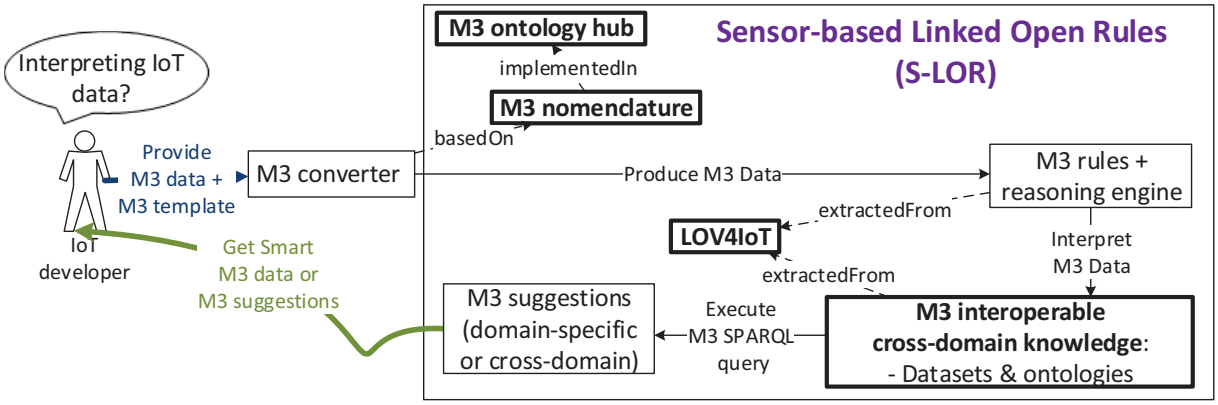

Figure 8. Assister les développeurs à interpréter les données de capteurs

La Figure 9 résume l'ensemble du processus Sensor-based Linked Open Rules (S-LOR) qui a été intégrée dans le système M3 [5]. Dans ce schema, une même mesure (température 38,7 DegC) est décrite dans deux domaines différentes: la santé pour le chemin A, les prévisions météorologiques pour le chemin B. Cet exemple met en évidence la nécessité d' : (1) ajouter explicitement la description des mesures du capteur, (2) interpréter les données des capteurs, et (3) combiner des domaines pour concevoir des applications inter-domaines. La première boîte, appelée « IoT data » récupère les données issues des capteurs avec des descriptions telles que température 38,7 DegC. Ces descriptions sont au format SenML, mais le système M3 pourrait

supporter d'autres formats. Puis, dans la deuxième boite, appelée «Semantic IoT data», les données sont sémantiquement annotées selon la nomenclature et l'ontologie M3, une étape qui est nécessaire pour les prochaines étapes. Puis, dans la quatrième boîte, appelée «Semantic Rule, new domain concept», l'approche S-LOR est exploitée, un ensemble de règles interopérables basées sur l'ontologie M3 et la nomenclature afin de déduire des abstractions de haut niveau. Dans le chemin A, S-LOR déduit la notion fièvre, alors que dans le chemin B, S-LOR déduit le concept chaud. Puis, dans les cases 4 et 5, appelés «Domain ontology» et « Domain dataset », les résultats du raisonnement fournis par S-LOR sont liés aux ontologies et à des bases de données de domaine interopérables. Puis, à l'étape 6, « Cross-domain applications », les connaissances du domaine interopérable M3 sont utilisées pour combiner les domaines et fournir des suggestions. Par exemple, nous avons lié des bases alimentaire aux symptômes de la fièvre dans la voie A, et des bases alimentaires liées à la saison dans le chemin B. Ainsi les bases alimentaires sont utilisées à la fois dans la santé et dans la météo ce qui permet d'entrecroiser facilement les connaissances et donc les domaines. Enfin, à l'étape 7, une demande SPARQL, un langage pour interroger les données de capteurs, interroge la base de connaissance inter-domaine interopérable pour obtenir des données et des suggestions intelligentes. Les résultats fournis sont ensuite exploités dans l'application finale telle que l'application naturopathie qui suggère des remèdes maison lorsqu'une fièvre est détectée.

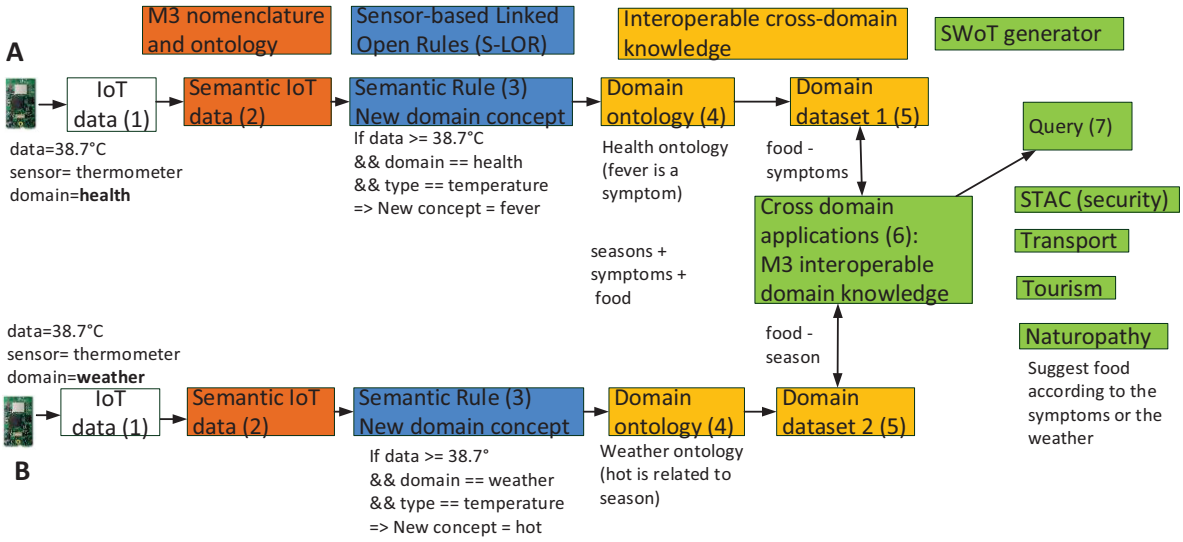

Figure 9. Le composant S-LOR intégré dans le système M3

4. LOV4IoT

L'outil nommé **Linked Open Vocabulaires for Internet of Things (LOV4IoT)** permet la réutilisation de l'expertise de la connaissance de domaine. Nous avons poursuivi une analyse extrêmement approfondie des connaissances de domaine liés aux capteurs pour répondre aux questions suivantes:

- Quels domaines utilisent des capteurs?
- Quelles ontologies existent pour chacun de ces domaines?
- Quelle raisonnement est utilisé dans chacun de ces domaines pour interpréter les données de capteurs?
- Est-ce que l'ontologie est réutilisable, par exemple téléchargeable depuis un site web?

- Quelles technologies ou outils sont utilisés pour mettre en œuvre l'ontologie ou les règles?
- Est-ce que l'ontologie respectent les bonnes pratiques du web sémantique?

Pour exploiter l'expertise de la connaissance du domaine et faciliter le développement d'applications IdO, nous avons conçu la base de connaissance LOV4IoT qui référence plus de 270 projets exploitant les technologies du web sémantique et les capteurs dans divers domaines tels que la santé, la domotique, l' alimentaire, l'agriculture, le tourisme, la sécurité, le transport et la ville intelligente. Nous avons étudié, identifié, référencé, analysé et extrait: (1) les capteurs et leurs mesures utilisés fréquemment, (2) les projets qui sont ré-exploitables dans d'autres domaines, par exemple, l'application naturopathie combine la santé, la météo et la cuisine intelligente, (3) les projets basés sur des ontologies, (4) les projets utilisant des systèmes à base de règles, (5) les experts du domaine ont publié leurs travaux dans des conférences scientifiques, (6) ils ont expliqué pourquoi ils utilisent les technologies sémantiques, (7), ils décrivent comment ils évaluent leur ontologies, et (8) le code de l'ontologie ou jeux de données peuvent être réutilisé dans de nouvelle applications inter-domaines.

5. STAC

L'outil **Security Toolbox : Attacks & Countermeasures (STAC)** est une base de connaissance de sécurité inter-domaine qui aide les développeurs et les concepteurs n'étant pas expert en sécurité à choisir les mécanismes de sécurités les plus adaptés à leur besoin pour sécuriser leurs applications ou leurs architectures IdO. Toutefois, l'outil STAC peut être utilisé pour sécuriser tout type d'applications.

STAC a été conçu en utilisant la même approche décrite dans M3 concernant la base de connaissance interopérable qui enrichir les données des capteurs. Le générateur STAC s'inspire également du générateur SWoT. En effet, à partir d'un capteur, nous pouvons filtrer les projets utilisant ce même capteur et incluant des mécanismes de sécurité pour sécuriser les données ou les applications.

STAC est une base de connaissance qui classifie de nombreuses technologies ainsi que leurs attaques et les mécanismes de sécurité existants ; mais aussi les propriétés de sécurité, les caractéristiques des mécanismes de sécurité (avantages et inconvénients), etc… De plus, STAC est une base de connaissance inter-domaine car elle couvre divers domaines de sécurité: les réseaux de capteurs, les cellulaire réseaux (2G, 3G, 4G), les réseaux sans fil (réseaux de capteurs, Wi-Fi, Bluetooth, RFID, Wimax, Zigbee, Manet, Mesh), les applications Web et l'administration réseaux. L'ontologie STAC et la base de donnée associée respectent les bonnes pratiques du web sémantique, sont publiées en ligne et ont été référencé par le Linked Open Vocabularies (LOV), un catalogue de vocabulaires maintenu par des experts en web sémantique qui intègre une nouvelle ontologie seulement lorsqu'elles respectent certains critères et est validée avec des outils sémantiques. En outre, le générateur STAC exploite cette base de connaissances de la sécurité pour aider les utilisateurs à la sécurisation des applications. La base de connaissance de sécurité STAC a été utilisé pour construire une application de sécurité inter-domaine où l'utilisateur peut naviguer d'un domaine à un autre (ex., réseaux de capteurs ou Bluetooth). Dans le futur, nous enrichiront automatiquement la base de connaissances de sécurité STAC avec les nouvelles technologies, les attaques et les mécanismes de sécurité. Nous souhaitons également améliorer la convivialité de l'interface graphique. Une autre étape sera d'intégrer automatiquement la demande pour fixer les mécanismes de sécurité (par exemple, le cryptage de

données avec des algorithmes cryptographiques) en générant le code requis. Prenant également inspiration du système de règles S-LOR, nous pourrions intégrer des règles de sécurité à STAC telles que « système actuellement vulnérable » ou encore « vérifier la robustesse de l'application ».

IV. Cas d'application du système M3

Le système M3 n'est pas exclusif à la communauté Internet des Objets. De nombreux acteurs peuvent bénéficier de M3 tels que les développeurs, les utilisateurs finaux, les experts du web sémantique, les experts du domaine, les experts en normalisation et les distributeurs de capteurs et boites intelligentes.

De plus, nous proposons cinq cas d'utilisation. Le premier cas d'utilisation concerne l'interface du système M3 avec les appareils mobiles Android. L'utilisation de M3 avec les appareils Android a montré que le système est flexible et réalisable également en dehors du 'Cloud'. Le deuxième cas d'utilisation démontre que le système M3 peut être intégré dans un tableau de bord de voiture. Le troisième cas d'utilisation démontre une approche centrée sur l'utilisateur final qui utilise M3 incorporé à un réfrigérateur intelligent. Le quatrième cas d'utilisation démontre une approche centrée sur l'utilisateur final qui utilise M3 intégré dans des bagages intelligents. Enfin, le cinquième cas d'utilisation est l'application de sécurité STAC, basée sur une base de connaissance de sécurité entrecroisant les domaines spécifiques à la sécurité tels que les réseaux de capteurs, les communications sans fils, les application web, l'administration réseau, etc… Cette base de connaissance inter-domaine peut être exploitée par les développeurs afin de choisir les mécanismes de sécurité les plus adaptés à leur besoins pour sécuriser leurs applications et architectures IdO.

1. Embarquons M3 dans les voitures intelligentes

Une application du système M3 serait de l'intégrer dans le tableau de bord des voitures. Cette application fournirait des suggestions au conducteur en fonction des conditions météorologiques afin d'actionner ou pas certains dispositifs (voir Figure 10). Ces applications sont possibles grâce à la phase de découverte des capteurs dans lequel un capteur de précipitation est intégré à la voiture intelligente est est automatiquement reconnu. Ensuite, une fois que les données générées par ce capteur ont été envoyé au système M3, qui peut facilement les interpréter et fournit des suggestions grâce au moteur de raisonnement S-LOR. Une telle application est conçu en téléchargeant le bon package en fonction des capteurs exploités dans la voiture. Le résultat de cette application est une interface conviviale qui propose d'allumer les feux de brouillard parce que S-LOR interprète qu'il pleut et fournit ces suggestions. La Figure 10 est une maquette afin de montrer une interface conviviale pour les utilisateurs finaux. Le véritable démonstration qui a été mis en œuvre sur le 'Cloud' peut être testée en ligne[14].

[14] http://www.sensormeasurement.appspot.com/?p=transport

Figure 10. M3 integré dans un tableau de bord de voiture

2. L'application naturopathie

La Figure 11 montre que M3 pourrait être intégré dans des réfrigérateurs intelligents. Par exemple, les utilisateurs prennent leur température du corps grâce à un capteur connecté au réfrigérateur. La température de 40 degrés Celsius est ainsi mesurée et ensuite interprétée automatiquement par le réfrigérateur qui déduit que l'utilisateur a de la fièvre. Le frigidaire exploite, le système Sensor-based Linked Open Rules (S-LOR), plus précisément le moteur de raisonnement et les règles M3 fournies pour en déduire de telles connaissances. La base de connaissance naturopathie a été intégrée dans le réfrigérateur, et contient les bases de données alimentaires et leurs relations avec des bases de données associées à la santé. Plus particulièrement, ces bases de connaissances contiennent les relations entre les remèdes de grand-mères et les symptômes tels que la fièvre. Enfin, le réfrigérateur suggère quelques remèdes grand-mères tels que le miel, le citron ou des infusions de thym pour aider les utilisateurs à lutter contre les microbes. Les utilisateurs peuvent faire confiance à de telles suggestions, puisque le réfrigérateur fournit même des informations concernant les raisons de ces suggestions [1].

La Figure 11 est une maquette d'une interface conviviale pour les utilisateurs finaux. La véritable démonstration a été mis en œuvre sur le 'Cloud', avec l'application « Naturopathie » qui peut être testée sur le Web[15] et est composée des sous-applications suivantes:

- Suggestion de remèdes de grand-mères en fonction de la température du corps.
- Suggestion des aliments selon la température extérieure.
- Déduction de l'humeur en fonction de la luminosité extérieure.
- Déduction de l'humeur ou de maladies en fonction du rythme cardiaque, de la conductance de la peau et de la pression artérielle.
- Proposer une recette en fonction de la nourriture disponible dans votre cuisine.

Pour construire ces applications, nous avons simulé des jeux de données représentant des mesures de capteurs car nous n'avons pas eu l'occasion d'exploiter des capteurs réels. Ces ensembles de données sont accessibles en ligne aussi. La base de données issues de capteurs de

[15] http://www.sensormeasurement.appspot.com/?p=naturopathy

santé[16] simule le rythme cardiaque, la température, la pression artérielle, le cholestérol et les mesures de conductance de la peau. La base de données issues de capteurs de santé météo[17] simule la luminosité, la température, la vitesse du vent, l'humidité et mesures de précipitation. La base de connaissances naturopathie a été conçue manuellement et a été inspirée par la connaissance de domaine (santé, aliments et sciences affectives) qui nous avons référencés, classifiés et synthétisés dans la base de connaissance LOV4IoT.

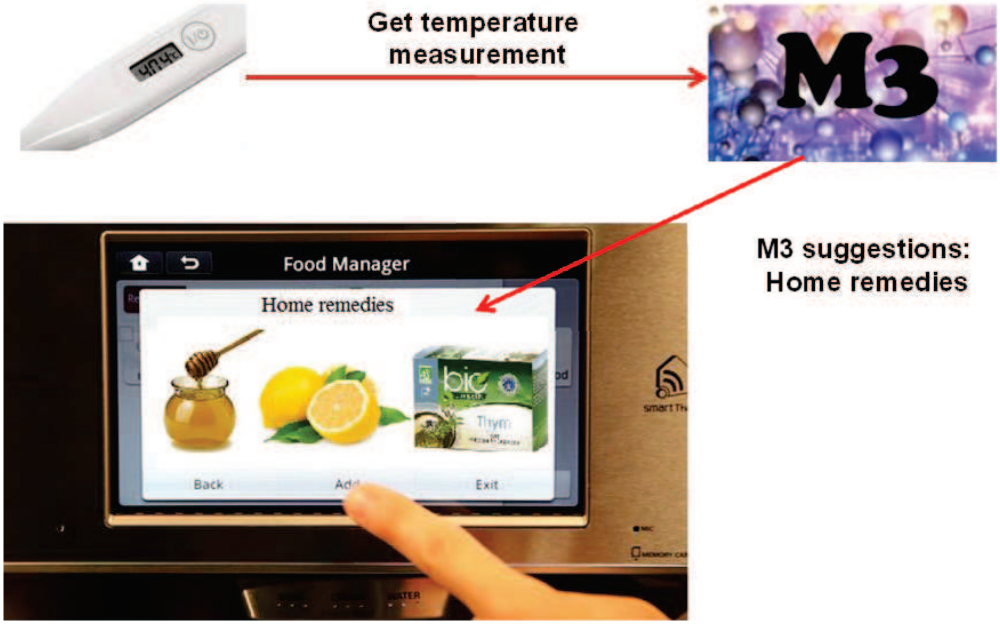

Figure 11. M3 intégré dans un frigidaire intelligent pour suggérer des aliments ou des remèdes de grand-mères

3. Le tourisme

Figure 12 montre que M3 pourrait être intégré dans des bagages intelligents. Par exemple, Nelly aime préparer ses vacances à la dernière minute et souvent bénéficie de réductions. Elle va sur le site LastMinute.com et choisit sa destination en fonction des offres du jour. Elle a trouvé une excellente destination avec un très forte réduction. Elle doit partir de chez elle dans deux heures, le système M3 est alors très efficace pour l'aider à préparer sa valise. M3 prend en compte la prévision météorologique ainsi que la destination, et suggère les vêtements appropriés pour ses vacances. Par exemple, pour aller dans un pays exotique, elle apportera un maillot de bain, des lunettes de soleil, un chapeau, etc. En même temps, Guillaume utilise également le site LastMinute.com pour aller à la montagne en hiver. M3 lui suggérera les vêtements et équipements nécessaires tels que gants, écharpe, col roulé, pull, etc.

[16] http://www.sensormeasurement.appspot.com/dataset/sensor_data/senml_m3_health_data.rdf
[17] http://www.sensormeasurement.appspot.com/dataset/sensor_data/weatherData_8KB_17Septembre2014.rdf

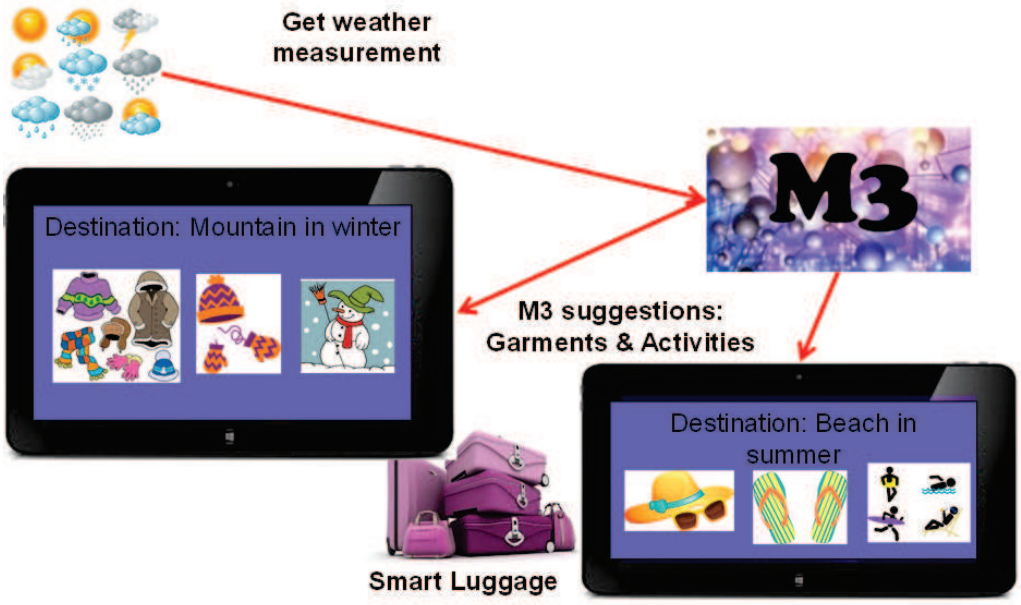

Figure 12. M3 intégré dans des bagages intelligents pour suggérer des vêtements et des activités

Figure 12 est une maquette affichant une interface conviviale pour les utilisateurs finaux. La véritable démonstration[18] a été mis en œuvre sur le 'Cloud' et est composé de quatre sous-applications:

- Proposant des activités en fonction de la météo.
- Proposer des vêtements en fonction de la météo.
- Suggérant vêtements quand il est enneigé.
- Suggérant des activités quand il est enneigé.

Pour construire ces applications, nous avons simulé des jeux de données de capteurs puisque nous ne avons pas eu l'occasion d'exploiter de vrais capteurs. Ces jeux de données sont accessibles en ligne aussi. Le jeux de données de météo[19] simule la luminosité, la température, la vitesse du vent, l'humidité et les mesures de précipitation.
Le jeux de donnée de neige[20] simule seulement deux mesures: la précipitation et la température. Ce jeux de données est principalement utilisé pour appliquer des règles plus complexes qui impliquent deux mesures en même temps.
La base de connaissances du tourisme a été repensé à la main et a été inspiré par les connaissances du domaine que nous avons référencé, classifié et synthétisé dans le catalogue de connaissance du domaine LOV4IoT.

V. Conclusion et directions futures

Nous concluons cette thèse en résumant nos contributions et en fournissant une perspective sur les orientations futures de cette recherche: (1) les défis à court terme pour améliorer notre projet Machine-to-Machine Measurement (M3), et (2) à long terme pour appliquer M3 dans d'autres domaines tels que la physique quantique et/ou les neurosciences. Enfin, nous

[18] http://www.sensormeasurement.appspot.com/?p=tourism
[19] http://www.sensormeasurement.appspot.com/dataset/sensor_data/weatherData_8KB_17Septembre2014.rdf
[20] http://www.sensormeasurement.appspot.com/dataset/sensor_data/snow_dataset.rdf

introduisons les impacts sociaux de cette thèse. Ce travail a été motivé par la nécessité de faire le traitement des données de capteurs interopérable, de combiner facilement des domaines hétérogènes et de construire des applications IdO encore plus intelligentes. Dans cette thèse, nous avons souligné le principal défi suivant: la combinaison de données issues de capteurs hétérogènes à l'aide des technologies du web sémantique afin de concevoir des applications IdO inter-domaines prometteuses. Ce défi a été divisé en 5 sous-défis qui nous expliquons ci-dessous (voir le Tableau 1):

- Défi A : L'interopérabilité des données IdO a été établie grâce au langage et à l'ontologie M3.
- Défi B : L'interprétation des données IdO a été résolu avec S-LOR en réutilisant les connaissances du domaine référencées dans LOV4IoT.
- Défi C : L'interopérabilité inter-domaine a été résolu avec les connaissances du domaine interopérable M3 et LOV4IoT.
- Défi D : La conception d'applications interopérables SWoT a été réalisée avec le système M3 et le générateur SWoT.
- Défi E : La sécurisation des applications IdO a été conçue avec STAC.

Ces défis ont été surmontés grâce aux contributions que nous expliquons ci-dessous.

Tableau 1. Défis mis en évidence dans l'état de l'art chapitre surmonté avec le système M3

Défis	Approches proposées
Défi A: Données IdO interopérables	Nomenclature et ontologie M3
Défi B: Interpréter les données IdO	S-LOR, LOV4IoT
Défi C: Interopérabilité des domaines et de la connaissance du domaine	Connaissance du domaine interopérable M3 + LOV4IoT
Défi D: Concevoir des applications IdO sémantiques	Générateur SWoT, Système M3
Défi E: Sécuriser les applications IdO	STAC

La première contribution est le système innovant **Machine-to-Machine Measurement (M3)** qui aide les développeurs dans la conception et la mise en œuvre d'applications IdO interopérables et inter-domaines. La principale nouveauté de M3 est de dissimuler les technologies du web sémantique aux développeurs. De plus, en utilisant M3, les machines peuvent automatiquement comprendre des informations de haut niveau et avec l'intelligence embarquée en eux, ils peuvent agir (actionneurs de contrôle, envoyer des notifications, etc.). M3 est composé du générateur SWoT qui produit une connaissance du domaine interopérable permettant de concevoir facilement des applications IdO basée sur la sémantique.

La seconde contribution est **Sensor-Based Linked Open Rules (S-LOR)** facilitant la réutilisation et la combinaison de règles interopérables afin de déduire des abstractions de haut niveau pour interpréter les données générées par les capteurs. S-LOR utilise le raisonnement logique basée sur des règles sémantiques compréhensibles à la fois par les humains et les machines. Les machines peuvent interpréter automatiquement les données IdO, et les fusionner afin de construire de nouvelles applications IdO. C'est une valeur significative comparée aux approches existantes qui sont fréquemment basées sur des techniques d'apprentissage automatique; en effet, notre solution permet de partager et réutiliser les règles interopérables

dans d'autres applications. Selon C. Perera et ses co-auteurs. les systèmes à base de règles ont moins de défauts que d'autres approches (ex., Apprentissage supervisé ou non supervisé, la logique floue) [8]. Notre approche proposée S-LOR peut surmonter les lacunes décrites dans [8]:"Les Règles doivent être écrites manuellement, ce qui peut être source d'erreurs et il n'y a aucune validation ou vérification de la qualité ". Cependant, grâce à notre approche innovante, ces limitations pourrait être facilement surmontées: (1) les règles sont conçus de manière interopérable afin d'être partagées et réutilisées, et (2) les règles pourraient être validées et notées par des experts du domaine. La connaissance du domaine interopérable M3 a été extraite du catalogue d'ontologies pertinent pour l'internet des objets que l'on a appelé « Linked Open Vocabularies for Internet of Things (LOV4IoT) ». LOV4IoT référence, synthétise et classifie plus de 200 projets basés sur des ontologies qui pourraient être ré-exploitées dans divers domaines tels que la santé, le transport, l'agriculture, la domotique, l'énergie intelligente, le tourisme, etc… Le système M3 a été intégré dans une architecture conforme aux standardisations tels ETSI M2M et est suffisamment générique pour être applicable dans différents scénarios tels que la naturopathie, le transport ou encore le tourisme. Ces scénarios ont été inspirés des scénarios proposés dans des projets européens tels que CityPulse[21] et IoT.est[22] ou encore ceux référencés dans le catalogue d'ontologies LOV4IoT. De plus, grâce à la souplesse et la maturité du système M3, ces scénarios peuvent être intégrés sur des plates-formes différentes telles que des téléphones mobiles, des boites intelligentes, ou le 'cloud'.

La troisième contribution est **Security Toolbox : Attacks & Countermeasures (STAC),** une nouvelle application de sécurité inter-domaines qui a été construite en utilisant la même approche que M3, mais dans le domaine de la sécurité. L'objectif de STAC est d'aider les développeurs non-experts en sécurité à sécuriser leur logiciel en suggérant les mécanismes de sécurité en fonction de leur besoins.

Bibliography

[1] Honey as complementary medicine: - a review, author=Sharma, Mukesh and Sharma, Deepak and Khan, Sheeba, year=2012, journal= International Journal of Pharma and Bio Sciences,.

[2] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. Semantics for the internet of things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(1):1–21, 2012.

[3] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[4] Amélie Gyrard and Christian Bonnet. A unified language to describe m2m/iot data, 2015.

[5] Amélie Gyrard, Christian Bonnet, and Karima Boudaoud. Enrich machine-to-machine data with semantic web technologies for cross-domain applications. In *WF-IOT 2014, World Forum on Internet of Things, 6-8 March 2014, Seoul, Korea*, Seoul, KOREA, REPUBLIC OF, 03 2014.

[6] Antonio J Jara, Alex C Olivieri, Yann Bocchi, Markus Jung, Wolfgang Kastner, and Antonio F Skarmeta. Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. *International Journal of Web and Grid Services*, 10(2):244–272, 2014.

[21] http://www.ict-citypulse.eu/scenarios/scenarios

[22] ict-iotest.eu/iotest/sites/default/files/files/public\%20deliverables/IoT.est_D2.1_V1.0.pdf

[7] ETSI M2M. Machine-to-Machine Communications (M2M); Study on Semantic support for M2M data, ETSI Techinal Report 101 584 v2.1.1 (2013-12), 2012.
[8] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *Communications Surveys Tutorials, IEEE*, 16(1):414–454, First 2014.
[9] Piyaporn Tumnark, Leandro Oliveira, and Nonchai Santibutr. Ontology-based personalized dietary recommendation for weightlifting. In *2013 International Workshop on Computer Science in Sports*. Atlantis Press, 2013.