Supporting the Context Life Cycle in Service-Oriented Computing

Presented By

Oscar Jair Cabrera Bejar

Advisors

Dr. Xavier Franch and Dr. Jordi Marco

Thesis

Submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy in Computing

UNIVERSITAT POLITÈCNICA de
C A T A L U N Y A
BARCELONA

Departament de Ciències de la Computació
Software and Service Engineering Group

May 2017

Acknowledgements

From the beginning to the end, undertaking this PhD has been a truly life-changing experience for me where I learned and acquired different skills, and taken advantage of unique opportunities in my life. This includes working on European projects and participating in important events and colaborations derived from the PhD. It would not have been possible to do without the support and guidance that I received from remarkable people.

Firstly, I would like to express my sincere gratitude to my advisors **Dr. Xavier Franch** and **Dr. Jordi Marco** for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my PhD study.

Besides my advisors, I would like to thank the **PhD committee, anonymous reviewers of the thesis and papers published during the PhD**, for their insightful comments and encouragement, which incented me to widen my research from various perspectives.

My sincere thanks also goes to all my friends and colleagues in the office: **Oscar Hernán, Lucia Mendez, Edith Zavala, Marc Oriol, David Ameller, Lidia López, Silverio Martínez, Cristina Palomares** and all the **GESSI** members, who helped me through all these years, shared many great and fun moments, and contributed with stimulating discussions that improved this work.

Last but not the least, I would like to thank **God** and my **family**: my parents and to my brother and sister for supporting me spiritually throughout this thesis and my life in general.

This work has been possible thanks to the funding of the National Council for Science and Technology (CONACyT), Spanish projects TIN2010-19130-C02-01, TIN2013-44641 and TIN2016-79269-R; and the SUPERSEDE project, funded by the European Union´s Information and Communication Technologies Programme (H2020) under grant agreement no 644018.

Abstract

Current software community players like academy and industry have been changing the traditional paradigms of software engineering towards context-awareness and distributed computing. Nowadays, service-oriented computing and context-aware computing are two emerging paradigms that are changing the way of designing, developing, providing and consuming software services. Whilst service-oriented computing is based on service-oriented architectures and it is focused on modelling functionality and providing flexible software services, context-aware computing is based on the context life cycle and it allows processing and changing the behaviour of such services given certain context information. The synergy between both paradigms is a core research topic in ubiquitous and pervasive computing widely applied to the Internet of Things and Smart Cities.

In the present PhD thesis, we exploit this synergy by focusing on context-aware computing from the perspective of service-oriented computing, which is also known as context-aware service-oriented computing. Such research topic involves the management of context within different essential phases of the context life cycle that show how the context data moves from phase to phase in software services within the paradigm of the service-oriented computing. Hence, the work done in this thesis involves different components and processes that have the aim to accomplish the context life cycle, namely the acquisition, modelling, reasoning and dissemination of the context in service-oriented computing. Particularly, we make an effort to provide both a context ontology for context modelling, context reasoning and high-level context dissemination, and a context-aware monitoring architecture for context acquisition and low-level context dissemination.

Such work of the thesis has been motivated for contributing in the solution of different issues mainly identified in the phases of context modelling and context acquisition that are a strong basis of the context life cycle. Firstly, in the context modelling we mainly identified the proliferation of several context models presenting some problems about: reusability, extensibility and adaptation. Secondly, in the context acquisition we mainly identified that existing monitoring infrastructures are not prepared to support the constant changes in their context and the context of other entities, including the services that they are supervising which provoke the provisioning of context data that is not reliable.

In summary, this thesis explores three big research questions: 1) What context data to acquire and to model? This involves the study of the current state of the art of context models, specifically: which are these proposals and how are they related, what are their structural characteristics, what context information is the most addressed, and what are their most consolidated definitions. 2) How to model context data? This involves the development of a three-level context ontology with the aim of improving the reusability, extensibility and adaptation capabilities of existing context models. 3) How to acquire context data? This involves the development of a context-aware monitoring architecture that can be easily configured, adapted or evolved according to the constant changes of the context.

The context model and the architecture proposed in this PhD thesis are validated through different scenarios and use cases, highlighting their integration in SUPERSEDE (www.supersede.eu), a European project in the H2020 program for fulfilling some requirements of data acquisition and management demonstrating that the context life cycle is supported.

Resum

Els actors actuals de la comunitat de software, com l'acadèmia o la indústria, han anat canviant els paradigmes tradicionals de l'enginyeria de software cap a la sensibilitat al context i la computació distribuïda. Avui dia, la computació orientada a serveis i la computació conscient del context són dos paradigmes emergents que estan canviant la forma de dissenyar, desenvolupar, proporcionar i consumir serveis de software. Mentre que la computació orientada a serveis es basa en arquitectures orientades a serveis i se centra en el modelatge de la funcionalitat i la prestació de serveis de software flexibles, la computació sensible al context es basa en el cicle de vida del context i permet el processament i canviar el comportament d'aquest tipus de serveis donada una determinada informació del context. La sinergia entre els dos paradigmes és un tema central de recerca a la computació ubiqua i omnipresent, àmpliament aplicada a la Internet de les coses i les ciutats intel·ligents.

En la present tesi doctoral explotem aquesta sinèrgia, centrant-se en la computació sensible al context des de la perspectiva de la computació orientada a serveis, que també es coneix com computació orientada a serveis sensibles al context. Tal tema de recerca implica la gestió de contexts en diferents fases essencials del cicle de vida del context que mostren com les dades de context es mouen d'una fase a l'altra en serveis de software dins del paradigma de la computació orientada a serveis. Per tant, el treball realitzat en aquesta tesi consisteix en diferents components i processos que tenen l'objectiu d'aconseguir el cicle de vida del context, és a dir, l'adquisició, el modelatge, el raonament i la difusió del context en computació orientada a serveis. En particular, fem un esforç per proporcionar tant una ontologia de context per a la modelització, raonament i difusió del context d'alt nivell, i una arquitectura de monitorització sensible al context per a l'adquisició i difusió del context de baix nivell.

Aquest treball de tesi ha estat motivat per contribuir a la solució dels diferents problemes identificats principalment en les fases de modelatge de context i adquisició de context que són una base sòlida del cicle de vida del context. En primer lloc, en el modelatge de context es van identificar principalment la proliferació de diversos models de context que presenten alguns problemes sobre: reutilització, l'extensibilitat i l'adaptació. En segon lloc, en l'adquisició del context identifiquem principalment que les infraestructures de monitorització existents no estan preparats per suportar els canvis constants en el seu context i el context d'altres entitats, incloent-hi els serveis que s'estan supervisant, que provoquen un aprovisionament de dades de context que no és fiable.

En resum, aquesta tesi explora tres grans preguntes de recerca: 1) Quines dades de context cal adquirir i modelar? Això implica l'estudi de l'estat actual de la tècnica dels models de context, en concret: ¿quines són aquestes propostes i com es relacionen, quines són les seves característiques estructurals, quina informació de context és la més adreçada, i quines són les seves definicions més consolidades. 2) Com modelar les dades de context? Això implica el desenvolupament d'una ontologia de context de tres nivells amb l'objectiu de millorar les capacitats de reutilització, extensibilitat i adaptació dels models de context existents. 3) Com adquirir dades de context? Això implica el desenvolupament d'una arquitectura de monitorització sensible al context que pot ser fàcilment configurat o adaptat d'acord amb els canvis del context.

El model de context i l'arquitectura proposada en aquesta tesi doctoral es validen a través de diferents escenaris i casos d'ús, destacant la seva integració en SUPERSEDE, un projecte europeu en el programa H2020 per al compliment d'alguns requisits d'adquisició i gestió de dades que demostra que es dóna suport al cicle de vida del context.

Table of Contents

List of Figures

List of Tables

"One, remember to look up at the stars and not down at your feet. Two, never give up work. Work gives you meaning and purpose and life is empty without it. Three, if you are lucky enough to find love, remember it is there and don't throw it away."

- Stephen Hawking

CHAPTER 1

Introduction

1.1 Scope and terminology

The present PhD thesis has been developed within the scope of context-aware computing and context management for accomplishing the context life cycle, i.e., acquisition, modelling, reasoning, and dissemination of the context. Specifically, we focus on context modelling and context acquisition for supporting the context life cycle. Such research topics are described in the following subsections in order to provide a general scope and clarify the concepts and terminology used in the thesis.

1.1.1 Context

The term *context* is widely used with very different meanings or very general definitions such as the definition given by the Merriam-Webster dictionary as *"the interrelated conditions in which something exists or occurs[1]"*. This definition is too general and therefore, it does not help to understand what a context is and how it can be effectively applied. Bazire and Brézillon presented an attempt to point out some problematic issues about the understanding of context, with more than 100 definitions analyzed, noting that its definition varies depending on the study field [1]. Similarly, Dey et al. evaluated 10 different context definitions, remarking the restrictions that each of them has for identifying new context [2]. Based on their study and considering its expressiveness for describing different entities through context allowing distinguishing between what is context and what is not, we adopt the context definition provided by Dey specified as follows:

"*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves****"*** *[3].*

This definition of context is very important, since it provides information about the present status of people, places, things and devices in the environment [4]. At this point, it is possible to deduce that context exists everywhere; its presence sometimes is explicit but possibly in a higher percentage also implicit. In the context life cycle is also considered the existence of two types of context, personal and generic [5]. In the first case, context affects only the entity to which it belongs (e.g. blood pressure only affects instances of the human class); in the second case, it affects more than one entity (e.g., humidity affects both, instances of human and device classes).

1.1.2 Context-aware computing

One of the major applications of context is in the context-aware computing research area. To avoid increasing complexity and allow the users to concentrate on their tasks, services must be aware of their context and automatically adapt when it changes; this ability is known as context-awareness [6]. One goal

[1] http://www.merriam-webster.com/

of context-aware systems is to acquire and utilize context information to provide services that are appropriate to particular people, place, time, event, etc. [7]. According to Byun and Cheverst, *"a system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context of use"* [8]. Similarly, Dey specifies that *"a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task"* [9].

Context-aware computing involves a *context life cycle* for context management. According to Perera et al. [10], there are four essential phases in context management systems. First, context acquisition, responsible for acquiring the context from different sources such as physical, virtual and logical sensors. Second, context modelling, responsible for modelling and representing the data acquired in meaningful context information. Third, context reasoning, aimed to derive high-level context information from low-level context raw data. Finally, context dissemination, aimed to exploit high-level and low-level context in services or other entities interested in this information. In the present thesis, our contribution is mainly focused on context acquisition and context modelling for supporting the context life cycle.

1.1.2.1 Context acquisition

Context acquisition is carried out by different monitoring tools or sensor types that can be integrated into a monitoring infrastructure. According to Perera et al. [10] *"a sensor is a system which provides information of an entity that can be divided into three categories: physical, virtual and logical"*. The physical ones are tangible sensors that retrieve information such as temperature, humidity, noise, etc. Virtual sensors can be offered as a software service to retrieve data that cannot be measurable physically such as email, chat, social networking data, quality of service, etc. Logical sensors combine the two previous types in order to produce more meaningful information.

The monitoring infrastructure is important during different stages of service-oriented computing such as service discovery where the monitoring system is useful to obtain the real quality of service (QoS) for a reliable service selection. This importance and the diversity of use has provoked the proliferation of different monitoring tools and solutions that have been developed and offered by several enterprises with or without profit, and by the contribution of the scientific research, suggesting a selection process to use the most appropriate monitoring tool for a particular need.

1.1.2.2 Context modelling

In the academic research, there exist relevant perspectives on context modeling. According to Henricksen, *"a context model identifies a concrete subset of the context that is realistically attainable from sensors, applications and users and able to be exploited in the execution of the task. The context model that is employed by a given context-aware application is usually explicitly specified by the application developer, but may evolve over time"* [11]. Context modeling should provide a well-defined structure of context information facilitating the activities carried out in all other context life cycle stages [10]. When the model does not allow representing relations and dependencies, it is difficult to carry out reasoning and inference to generate further context information classes. Furthermore, the raw data acquired from sensors must be translated into context information classes to give sense and structure. In this regard, context modeling is an effective method of gathering, representing and sharing context information across different information systems [12].

Context information represents the basic element in a context model adopting atomic items that can be classified based on different categorization and formalization schemes following a systematic methodology. As stated by Crowley et al., *"context is an information space that can be modelled as a directed state graph, where each node denotes a context, and edges denote the conditions for change in* *context"* [13]. Each context is defined by a set of entities (typically including literal values, as well as real-world and information objects), a set of roles (for example, functions) that entities may satisfy, a set of relations between the entities, and a set of situations. Entities, roles, and relations are modelled as expressions over observables captured and inferred by the system at the appropriate level of abstraction. A well-defined context model will minimize the complexity of context-aware apps, enhancing their maintainability and ability to evolve [14].

Different approaches for modelling context can be found in the literature. Recently, Perera et al. [10] presented a comparison of the six most popular context modeling techniques namely *key-value*, *mark-up*, *graphical*, *object*, *logic* and *ontology*. The analysis and evaluation of the previous techniques for context modelling given by Perera et al. [10] and Strang and Linnhoff-Popien [15] indicate that the most appropriate technique to manage context is the ontology-based modelling. According to Sudhana et al. [16], the main purpose of ontology-based context models is to enable semantic interoperability and to provide common understanding of the structure of context information among users. Ontologies are believed to be a key feature in the making of context-aware distributed systems because they support knowledge sharing, reasoning and interoperability [17][18][19].

For all these reasons, we have adopted the *ontology-based modelling* approach for representing, reasoning and disseminating context.

1.1.3 Service-oriented computing from the context perspective

According to Papazoglou et al. *"service-oriented computing (SOC) is a paradigm that uses services as reference point to support the development of rapid, low-cost, interoperable, evolvable, and massively distributed applications"* [20]. From their perspective, services perform functions that range from answering simple requests to executing sophisticated business processes requiring peer-to-peer relationships among multiple layers of service consumers and providers.

In today's world, there are different kinds of services created to facilitate the life of humans in their daily tasks, i.e., services that have been developed to solve different needs according to certain requirements of different human contexts. As a result, an enormous explosion in the offering of services has occurred; in fact, it can be observed that for a given need, a wide range of services can be found. According to Bon, there is a growth in consumer services driven by various social, economic and technology factors such as rising demand for social services, size and role of the public sector, complexity of work environments, etc. [21]. The key of services lies in the value offered as the following definition specifies: *"a service is a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks"* [21].

According to Badidi and Taleb context information is useful to identify various situational circumstances from the viewpoint of services [22], such as:

- The location and identity of the client who invoked the service (whether it is a person or another service).
- The time at which the client invokes the service.
- The activity that the client is carrying out at the time it invokes the service.
- The preferences that the client may have defined prior to invoking the service.
- The security and privacy policies associated with the client of this service.
- The device (laptop, PDA, smartphone, etc.) that the client is using to invoke the service.

In common services, it is important to take into account this context information. A good example is given in [23] describing a restaurant service. In this case, it is argued that if context information is not provided, the restaurant would typically show a single menu all day long. Instead, if context information is provided, the restaurant could eventually show different suggestions depending on who is walking by. If parents with children walk by, the restaurant would show the children's menu; if a couple is looking at it in the evening, it would show the menu for a candle light dinner; and if it is hot and sunny in the afternoon, the restaurant would advertise the selection of ice cream.

1.2 Motivation

The adoption of context-aware computing as paradigm involves a context life cycle for context management as depicted in Fig. 1 from which lies the motivation of this thesis. Specifically, in acquisition and modelling of context, which are the basis stages for the remaining two, i.e., context reasoning and context dissemination.

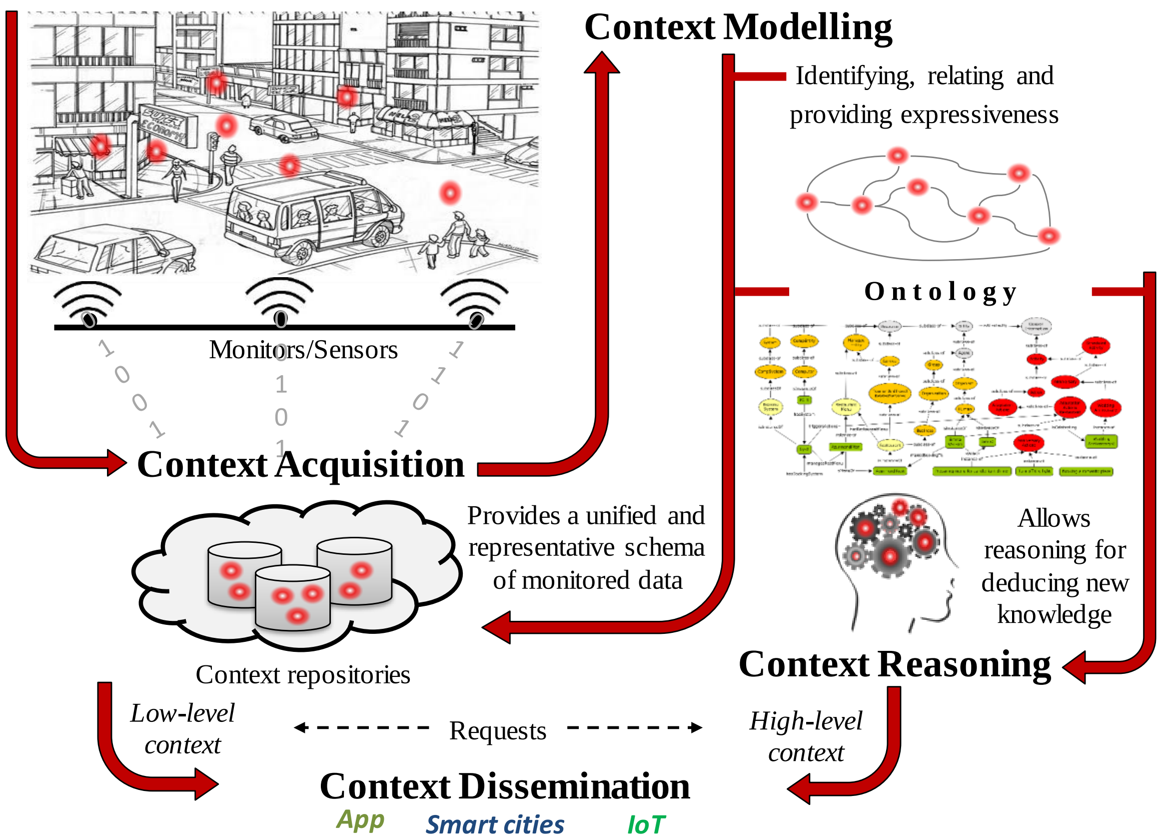


Fig. 1 Context life cycle

Based on the current research challenges for supporting the context life cycle through the acquisition and modelling of context, this thesis is motivated by the need of contributing in solving the following research gaps that were identified through a state of the art:

- *For context acquisition.* Context acquisition plays an important role in the context life cycle since it is responsible for the acquisition and dissemination of low-level context data (see Fig. 1). To acquire and provide reliable context data that can be later processed, a monitoring infrastructure should maintain the functionality and quality levels desired. For this purpose, this infrastructure should support constant changes in its context and in the context of other entities, including the services that it is supervising. However, several research challenges in the field of monitoring infrastructures supporting capabilities of reconfiguration, evolution and adaptation to face the constant context changes are still open which provoke the provisioning of not reliable context data.

- *For context modelling.* Context modelling is responsible of providing a unified and representative schema of monitored data to be stored in a consistent way, for deducing new knowledge, and for providing high-level context data (see Fig. 1). Context modelling has emerged suggesting different formalisms to structure concepts and definitions of context information. However, as it happens in many other areas, it does not exist a single model agreed by the scientific community; instead several proposals have been presented for specific or general purposes. These context modelling proposals may diverge in several matters: facets addressed, size and completeness, structure and terminology, underlying principles, semantic factors, type of artifact chosen for knowledge representation (e.g. object-based, ontology-based), etc. Whereby, it may be argued that it is necessary an effort to contribute solving these open issues. Furthermore, although there are different context models proposed in the literature, there exists other important gaps that are not fully covered, particularly related to the reusability of the models themselves and the lack of unified and consolidated the context knowledge. On the one hand, it seems that most of context-aware computing proposals prefer to design their own customized context model instead of reusing an existing one. The main reasons for this behavior is that current context models present some big problems concerning: reusability, extensibility and adaptation. On the other hand, the main cause identified is related to the lack of homogeneity of the elements provided among the contributions, as well as the shortage of their definitions. This problem calls for efforts to consolidate the context knowledge already available and to specify a clear schema of knowledge reutilization, which also is almost non-existent in the proposed context models.

1.3 Objective and research questions

The objectives of the present thesis are defined to cope the research gaps previously specified by means of:

- To investigate on the definition and structure of different context models based on ontologies, and provide a framework of common understanding for the definition of **what to acquire and to model**. The research question to achieve this objective is formulated in RQ1 (see Table 1).
- To investigate on the most adequate context modelling approach for supporting the activities of **how to model** context knowledge that can be easily reused, extended or adapted, and develop a model satisfying such capabilities. The research question to achieve this objective is formulated in RQ2 (see Table 1).
- To investigate on the different dynamic capabilities of existing monitoring infrastructures for supporting the activities of **how to acquire** context through an infrastructure that can incorporate reconfiguration, evolution and adaptation capabilities to face the constant changes of context, and develop a context-aware monitoring architecture that satisfies such capabilities. The research question to achieve this objective is formulated in RQ3 (see Table 1).

Table 1. Research questions

ID	Research question
RQ1	In the field of service-oriented computing, do ontology-based context models proposed so far provide an adequate and structured set of context knowledge pieces to be reused in prospective proposals that can define their context?
RQ2	In the field of service-oriented computing, how to model, unify and consolidate the context knowledge of existing contributions for improving their reusability, extensibility and adaptation capabilities?
RQ3	In the field of service-oriented computing, how to acquire context through a complex context-aware monitoring system with the required reconfiguration and adaptation capabilities to satisfy the requirements in constantly changing situations?

1.4 Methodological approach

This section addresses the methodological approach adopted for characterizing the research strategy conducted in the present thesis. It is based on a methodology that provides different classifications to help characterize the software engineering research provided by Shaw in [24]. Such classifications include research setting, research product and validation techniques that are described and characterized below according to the research questions of the thesis previously defined.

1.4.1 Research setting

The research setting classification allows to investigate and formulate the problem, i.e., the problem is transferred into a research setting with the aim of finding solutions to it. For this purpose, Shaw proposes five research setting types with some kinds of questions posed by each that are specified as follows:

- *Feasibility*. Is there an X, and what is it? Is it possible to accomplish X at all?
- *Characterization*. What are the important characteristics of X? What is X like? What, exactly, do we mean by X? What are the varieties of X, and how are they related?
- *Method/Means*. How can we accomplish X? What is a better way to accomplish X? How can I automate doing X?
- *Generalization*. Is X always true of Y? Given X, what will Y be?
- *Selection*. How do I decide between X and Y?

From this perspective, the research setting types that better fit the research presented in the thesis are *characterization* for RQ1 and *method/means* for RQ2 and RQ3, since:

→ In the research question 1 (**RQ1**, see Table 1), the context knowledge from existing context models based on ontologies is characterized by studying its characteristics, varieties, definitions and relations.

→ In the research question 2 (**RQ2**, see Table 1), we define the means on how to model this context knowledge for accomplishing its reusability, extensibility and adaptation.

→ In the research question 3 (**RQ3**, see Table 1), we define the means on how to acquire this context knowledge being context-aware for supporting reconfiguration, evolution and adaptation.

1.4.2 Research product

The research product classification represents the main outcomes of the research, i.e., allows specifying the tangible research results of the research. For this purpose, Shaw proposes five research product types described as follows:

- *Qualitative or descriptive model*. Organize & report interesting observations about the world. Create & defend generalizations from real examples. Structure a problem area; formulate the right questions. Do a careful analysis of a system or its development.
- *Technique*. Invent new ways to do some tasks, including procedures and implementation techniques. Develop a technique to choose among alternatives.
- *System*. Embody result in a system, using the system development as both source of insight and carrier of results.
- *Empirical predictive model*. Develop predictive models from observed data.
- *Analytic model*. Develop structural (quantitative or symbolic) models that permit formal analysis.

From this perspective, the research products of the thesis are *qualitative or descriptive model* for RQ1, and *technique* for RQ2 and RQ3, since:

→ In the research question 1 (**RQ1**, see Table 1), it is organized, analyzed and reported observations through systematic questions about the context knowledge of existing context models based on ontologies.
→ In the research question 2 (**RQ2**, see Table 1), it is provided a context model for improving the way to do tasks of reusing, extending and adapting context knowledge.
→ In the research question 3 (**RQ3**, see Table 1), it is provided a context-aware monitoring architecture for context acquisition from which is improved the way to do tasks of reconfiguration, evolution and adaptation.

1.4.3 Validation techniques

Finally, the validation techniques allow validate the research results to demonstrate that they satisfy the research setting previously specified. For this purpose, Shaw proposes five validation techniques and their characters of validation described as follows:

- *Persuasion*. Validation by means of a technique, design or example.
- *Implementation*. Validation by means of the implementation of a system or technique.
- *Evaluation*. Validation with respect to a descriptive model, a qualitative model or an empirical quantitative model.
- *Analysis*. Validation by means of an analytic formal model or an empirical predictive model.
- *Experience*. Validation expressed in a qualitative or descriptive model, as decision criteria or an empirical predictive model.

From this perspective, the validation techniques adopted for validating the results of the thesis are *persuasion* that in general is applied along the thesis by providing examples that clarify how a proposed idea works; *implementation* for RQ2 and RQ3, *evaluation* for RQ1, RQ2 and RQ3; and *experience* for RQ2 and RQ3, since:

→ In the research question 1 (**RQ1**, see Table 1), the evaluation is conducted with respect to a common classification of validity concerns and defined criteria.
→ In the research question 2 (**RQ2**, see Table 1), the evaluation is conducted in three ways. First, by reusing, extending and adapting the proposed context model through descriptive scenarios and defined criteria. Second, by instantiating the proposed context model with respect to defined criteria and requirements of internal components of a European project. Third, by integrating the proposed context model in a conceptual context-aware framework architecture for representing and describing its role in the context management according to basic requirements that satisfy the context life cycle. The validation through the implementation technique is carried out by developing the context model and by triggering queries from the perspective of descriptive scenarios demonstrating consistency and the required functionality of the context model in the context life cycle, i.e., exploitation and reasoning of context. Finally, the experience technique is conducted by integrating the context model in a real environment, in this case for fulfilling tasks of a European project.
→ In the research question 3 (**RQ3**, see Table 1), the evaluation is conducted in two ways. First, by instantiating the proposed context-aware monitoring architecture with respect to defined requirements and criteria of real use cases of a European project. Second, by integrating the proposed context-aware monitoring architecture in a conceptual context-aware framework for representing and describing its role in the context management according to the basic requirements that satisfy the context life cycle, i.e., acquisition and exploitation of context. The validation through the implementation technique is carried out by developing a prototype that demonstrate how can be managed the proposed context-aware monitoring architecture. Finally, the experience technique is conducted as RQ2.

1.5 Thesis contributions

The thesis contributions and scientific dissemination of the present thesis are divided as follows: mapping study of context models based on ontologies (RQ1), context model (RQ2) and context-aware architecture (RQ3). The specifications of such contributions are given in the remainder of this section.

1.5.1 RQ1 contributions

Context modelling plays an important role in the context life cycle for representing, reasoning and providing meaning to the collected context data (see Fig. 1). The main contributions of this research question are given by a systematic mapping on context modelling based on ontologies that attempts to provide a reference for prospective researchers and practitioners in the field, especially to help avoiding the definition of new proposals that do not align with current research. Specifically, the study evaluates the current state of the art of context models based on ontologies investigating the following aspects: (1) which are these proposals and how are they related; (2) what are their structural characteristics; (3) what context information is the most addressed; and (4) what are their most consolidated definitions. From this research perspective, specific contributions of the systematic mapping are pointed out as follows:

- Distribution of the context models based on ontologies along the time and the identification of their relationships.
- Size correlated with the number of context classes and levels of the context models based on ontologies.
- Coverage of the context knowledge definitions provided in the context models based on ontologies.
- Most addressed context information over the context models based on ontologies.
- Most consolidated definitions of context information.
- Consolidated taxonomy for organizing the most relevant classes of entities and context information found.

The contribution of this survey is to make available a unified and consolidated body of knowledge on context for service-oriented computing that could be instantiated and used as starting point in a variety of use cases including reconfiguration, adaptation and evolution of services. This sweeping view on the anatomy of context models based on ontologies may help to reuse and to understand the structure and semantic of existing context knowledge.

The scientific dissemination of the contributions of the systematic mapping study has been accepted to the SCI-indexed journal *Data and Knowledge Engineering* (I.F.2015: 1.500) [25].

1.5.2 RQ2 contributions

Although a lot of research has been made in the field of context modelling, most of the context-aware computing proposals prefer to design their own customized context model instead of reusing an existing one. The main reasons for this behavior is that current context models present some problems concerning reusability, extensibility and adaptation. To contribute solving these issues, the main contributions of this research question is a three-level context ontology for context modelling that can be easily reused, extended and adapted for specific or generic purposes. Such ontology consolidates the context knowledge already available from a modular perspective yielding a clear schema of knowledge reutilization. To do so, we gathered context resources of different ontologies to be integrated into unified and well-defined modules. Hence, the contributions of this research question have the aim to unify and consolidate a body of context knowledge that can be considered as benchmark in the context-aware computing facilitating the tasks of capturing, managing and distributing context and therefore, improving the value delivered by services. The proposed ontology comprises three levels of abstraction briefly described as follows:

- The upper-level constructed on the basis of the consolidated context taxonomy coming from the results of RQ1. Such upper-level has the aim of establishing a basic taxonomy of high-level context classes well-suited for reusing and extending the model.
- The middle-level has the aim of unifying and consolidating the ontological resources derived from RQ1 that provides an exhaustive study and analysis of existing contributions in context modelling. It acts as a bridge between the upper and lower levels with the aim of extending the upper-level ontology based on a prescribed process, and providing the resources required to be extended by domain-specific ontologies.
- The lower-level defines domain-specific ontologies which state a set of detailed classes highly dependent on the domain.

The scientific dissemination of the contributions of this research question was published in four venues: IEEE International Conference on Research Challenges in Information Science (RCIS, CORE-B[2]) [26], International Conference on Conceptual Modeling (ER, classified as CORE-A, in the form of short paper) [27], deliverable of the SUPERSEDE H2020 European project [28] and SCI-indexed journal *Software and Systems Modelling* (I.F.2015: 0.990) [29].

1.5.3 RQ3 contributions

Context acquisition plays an important role in the context life cycle, as it is responsible for the acquisition and dissemination of context data (see Fig. 1). To acquire and provide reliable context data that can be later processed, a monitoring infrastructure with the desired functionality and quality levels is needed. For this purpose, this infrastructure should support the constant changes in its context and the context of other entities, including the services that it is supervising. Hence, the main contribution of this research question is a context-aware monitoring architecture that is based on state-of-the art software patterns of service-oriented computing, enabling to plug and play different monitors that can be easily reconfigured, evolved or adapted for supporting highly dynamic environments.

The contributions of RQ3 also involve the study of several monitoring methods and techniques that were studied through a state of the art to support decision-making in the evolution and adaptation of software services ultimately leading to improved end-users' quality of experience (QoE). This vision foresees to exploit runtime contextual data gathered via monitoring (e.g., user events) for decision making. Furthermore, since the proposed context-aware architecture in some way will involve the selection of monitoring tools that can be integrated in such architecture, a quality model for analyzing monitoring tools is provided. At the moment, the contribution of this quality model is focused on commercial Web service monitoring tools that can be extended in a future work.

The scientific dissemination of the contributions of this research question were published in two venues: deliverables of the SUPERSEDE project [30][31] and the IEEE International Conference on Research Challenges in Information Science (RCIS, CORE-B) [32], and submitted to the SCI-indexed journal *Expert Systems with Applications* (I.F.2015: 2.981) [33].

1.5.4 List of publications

During the PhD thesis research different types of publications were conducted to contribute solving the research questions previously specified. Likewise, we worked in other research topics of the service-oriented computing namely QoS monitoring and Web service selection that although are not fully related to solve the research questions of the thesis, are used as aggregated cases of validation and future work of

[2] http://www.core.edu.au/conference-portal

the main results of the present thesis. Such research is presented as a list of publications in Table 2. Papers currently under revision in international journals are also presented.

Table 2. List of publications

Ref.	**Venue** [type]	**Title**	**Year**
	Submitted		
	SCI-indexed journals		
[33]	Expert Systems with Applications (ESWA) [*Journal*, I.F.2015: 2.981]	CAMA: A Context-Aware Monitoring Architecture for Highly Dynamic Environments	N/A
	Published		
	SCI-indexed journals		
[25]	Data & Knowledge Engineering (DKE) [*Journal*, I.F.2015: 1.500]	Ontology-Based Context Modelling in Service-Oriented Computing: A Systematic Mapping	2017
[29]	Software and Systems Modeling (SoSyM) [*Journal*, I.F.2015: 0.990]	3LConOnt: A Three-Level Ontology for Context Modelling in Context-Aware Computing	2017
	Conferences		
[27]	International Conference on Conceptual Modeling (ER) [*Int. Conference*, CORE: A]	A Middle-Level Ontology for Context Modelling (short paper)	2015
[26]	IEEE International Conference on Research Challenges in Information Science (RCIS) [*Int. Conference*, CORE: B]	A Context Ontology for Service Provisioning and Consumption	2014
[32]	IEEE International Conference on Research Challenges in Information Science (RCIS) [*Int. Conference*, CORE: B]	A Quality Model for Analyzing Web Service Monitoring Tools	2012
	Technical reports (not peer reviewed)		
[28]	SUPERSEDE [*H2020 Project* Deliverable]	D2.2: Data Management	2016
[31]	SUPERSEDE [*H2020 Project* Deliverable]	D1.4: Comprehensive Monitoring Techniques	2016
[30]	SUPERSEDE [*H2020 Project* Deliverable]	D1.1: Feedback Management and Monitoring Approaches	2015
	Other publications		
[34]	Computación y Sistemas (CyS)[3] [*Journal*, I.F.2013: 0.184]	Open Framework for Web Service Selection Using Multimodal and Configurable Techniques	2014
[35]	arXiv [*Technical Report (not peer reviewed)*]	WeSSQoS: A Configurable SOA System for Quality-Aware Web Service Selection	2011
[36]	Jornadas Científico-Técnicas en Servicios Web y SOA (JSWEB) [*National Conference*]	WeSSQoS: Un sistema SOA para la Selección de Servicios Web Según su Calidad	2009

[3] The journal appeared in Thomson Master Journal List during May 2013-April 2014 using print ISSN 1405-5546. At present, it is 1) indexed by Scopus, 2) part of the index of excellence of conacyt, and 3) part of Thomson Reuters Web of Science (SciELO collection). http://www.cys.cic.ipn.mx/ojs/index.php/CyS

1.6 Structure of the thesis

The structure of the present PhD thesis is organized in chapters that have the aim to provide an abstract of the most important findings and contributions of the thesis. Each chapter is oriented to cope the research questions exposed in Section 1.3 and their related contributions exposed in Section 1.5. Therefore, Chapter 2 entitle “What to acquire and to model: study on context modelling” refers to RQ1. Chapter 3 entitled “How to model: building a context model” refers to RQ2. Chapter 4 entitled “How to acquire: building a monitoring architecture” refers to RQ3. Chapter 5 entitled “Supporting the context life cycle: closing the loop”, refers to the integration of the results of RQ2 and RQ3. Finally, Chapter 6 provides the conclusions and future work of this PhD dissertation. Table 3 specifies the overview of each chapter.

Table 3. Chapters overview

Chapter	Overview
1	Provides the fundamentals and research settings of this PhD dissertation
2	It is divided in five main sections. Section 1 introduces systematic studies. Section 2 provides the planning for the review. Then, Section 3 reports the results of the review. Finally, Section 4 and 5 provide the discussions and threats to validity of the review, respectively.
3	It is divided in four main sections. Section 1 provides the methods and techniques employed for building the proposed model. Then, Section 2 provides the specifications of the proposed model. Section 3 applies such methods and techniques and presents the proposed model. Finally, Section 4 provides the validation of the proposal.
4	It is divided in six main sections. Section 1 specifies subresearch questions of RQ3. Section 2 provides a state of the art of context-aware monitoring infrastructures. Then, Section 3 presents the proposed context-aware monitoring architecture. Section 4 presents two instantiations of the proposed architecture in a form of proof of concept. Finally, Section 5 and 6 provide the implementation and validation of the proposed architecture, respectively.
5	It is divided in two main sections. Section 1 presents the proposed context-aware framework that integrates the context model and monitoring architecture proposed in the thesis for supporting the context life cycle. Finally, Section 2 provides the validations of the framework.
6	Provides the conclusions and future work of this PhD dissertation
7	Provides the references of this PhD dissertation

"If we knew what it was we were doing, it would not be called research, would it?"

- Albert Einstein

CHAPTER 2

What to acquire and to model: study on context modelling

Context-aware computing is critical in services that support different tasks of everyday's life. Context modelling, which plays an important role into the context life cycle, is an expanding research field and relevant issue into context-aware computing. The analysis and evaluation of context modelling issues becomes crucial for the value of services, which continuously need to evolve to be aware of the context that is responsible for their behavior.

Context models are thus the engineering formalism that have emerged to structure and conceptualize the concepts and definitions of the context knowledge. However, as it happens in many other software engineering research areas, there does not exist a single context model agreed by the scientific community; instead several proposals have been presented for specific or general purposes. These proposals may diverge in several matters: facets addressed, size and completeness provided, structure and terminology, underlying principles, semantic factors, etc. and even in the formalization degree for knowledge representation: basic classification (hierarchy or taxonomy), high-expressiveness (ontology), … Therefore, it may be argued that it is necessary an effort to assess the current state of the art of context modeling focused on analyzing and evaluating the context knowledge represented in existing contributions.

The aim of this first contribution of the thesis is to identify and relate existing proposals of context modelling based on ontologies, assess them with respect to some criteria and conclude with the most consolidated ontology-based context knowledge and the most remarkable gaps to bridge. As a result of such review, we have been able to identify the strengths and weaknesses of the current state of the art in context modelling based on ontologies. The importance of this contribution of the thesis lies mainly along two lines, namely providing an overview of existing context models and making available a consolidated context knowledge easy to reuse, specifically in current areas moving towards the innovation and deployment of new services such as Smart Cities, the Internet of Things and Self-Adaptive Systems.

2.1 Introduction to systematic literature studies

To conduct any type of literature study in an accurate and objective manner, it is necessary to use a precise and rigorous methodology. For such a purpose, the review on context modelling presented in this chapter uses the principles and guidelines proposed by Petersen et al. [37] and Kitchenham and Charters [38]. These guidelines have been derived from other existing studies used by medical researchers and adapted to reflect the specific problems of software engineering research. Since their inception, these guidelines have been widely used by software engineering researchers and when applied properly, they drastically reduce the risk of bias and incompleteness in the review results. The stages of the methodology are the following:

- **Planning the review:** activities performed before conducting the review. These activities include the identification of the need for a review, the statement of the research questions that the review is intended to answer, and the definition of the protocol that specifies the criteria that will be used to perform the review (e.g. search keywords, bibliographic databases, selection criteria, etc.).
- **Conducting the review:** activities that constitute the execution of the review and follow the protocol as defined in the previous phase. These activities include the identification of research, the selection of primary studies, the study of quality assessment, data extraction and synthesis.
- **Reporting the review:** activities to report the results of the review. It includes the specification of a dissemination mechanism, the format of the report, and its evaluation.

This methodology may be applied to the two main existing types of systematic literature studies, namely systematic literature reviews (SLR) [39] and systematic mappings (SM) [40]. Whilst SLRs investigate in detail some insightful research question, a systematic mapping is a method to review, classify, and structure papers related to a specific topic. The goal of a systematic mapping is to obtain an overview of existing approaches, outlining the coverage of the research field and its gaps. Given the objective of our study, it can be classified as a systematic mapping.

2.2 Planning the systematic mapping

This section reports the activities performed before conducting the review.

2.2.1 Identification of the need for a review

Petersen et al. [37] highlight the importance of motivating the need and relevance of any SM. For this purpose, this activity followed the recommendations given by Kitchenham and Charters [38] since they provide more evidence at this stage. Hence, researchers are required to seek and identify existing works related with the subject and assess them with respect to some criteria. Therefore, we searched for other reviews and assessed them by considering the relevant aspects of this review in contrast with the existing contributions to ensure and confirm that a new one is necessary. In order to expand the number of results retrieved, besides searching systematic literature studies, we also sought other studies published in scientific venues such as state of the art contributions, surveys, etc. To do so, we defined an additional search protocol, similar to the main search in the SM, to recognize such other reviews.

Therefore, we applied basically the protocol that will be explained in detail in the rest of the section, just adding to the search keywords the following terms: “state of the art”, “survey”, “review”, “SLR” and “systematic mapping”. As a result of applying automatic and manual searches, we found 16 papers fulfilling the search criteria. However, after inspecting them, we found that although these studies provide strong contributions in the fields of context and context-awareness they are oriented to other purposes that did not affect the objectives of this review.

In Table 4 we highlight the research directions covered by other systematic studies and based on them we differentiate our contributions. The criteria used for this purpose are described as follows:

- *Domain* is used to specify the stages of the context life cycle (CLF) addressed in the proposals: context acquisition (CA), context modelling (CM), context reasoning (CR) and context distribution (CD). We use the term "C-aware" in works that do not focus their contributions in a particular stage, specifying the stages more addressed or for which at least an overview is provided ([stage]).
- *Study object* refers to the study sample analyzed and evaluated in the proposals.
- *Context information* represents the contributions made in the analysis and evaluation of context information classes provided in proposed context models (e.g. sematic constrains among vocabulary). We use the term "limited" in the following situations: the contribution does not focus on this aspect but provides an overview on the matter ([overview]), it is focused on a specific context entity ([specific, entity]) or the information surveyed is limited ([information]).
- *Literature reviewed* refers to the number of projects, works or other artefacts selected in certain period of time to be analyzed and evaluated.
- *Result* specifies the main result obtained based on the *study object* criteria.

Table 4. Systematic Studies in Context Research

Proposal	Domain CLF	Study object	Context inf.	Literature reviewed	Result
Perera et al. [10]	C-aware [all stages]	Different solutions in context awareness from an IoT perspective	Limited [information]	50 ***2001-2011*** [projects]	Consolidate past research applied to IoT
Bettini et al. [14]	CM, CR	Requirements in context modelling and reasoning techniques in pervasive computing	Limited [information]	±60 ***1990-2009*** [techniques]	State-of-the-art in context modelling and reasoning
Bellavista et al. [41]	CD	Comparison of context data distribution solutions for mobile ubiquitous systems	Limited [overview]	37 ***2000-2010*** [solutions]	Unified architect-tural model and a taxonomy for CD
Strang and Linnhoff -Popien [15]	CM	Most important context modelling approaches	Limited [overview]	6 ***1993-2004*** [approaches]	Most appropriate model to represent context [ontologies]
Chen and Kotz [42]	C-aware [CA, CM, CD]	Context-aware applications, what context they use and how it is leveraged	Limited [specific, app context]	12 ***1992-2000*** [context-aware applications]	Context inf. used is limited and early managed
Baldauf et al. [43]	C-aware [CM, CD]	Middleware and server-based approaches specifically in their usability in dist. systems	Limited [overview]	8 ***1999-2005*** [architectures]	Strict division of the context data acquisition & use
Bolchini et al. [44]	CM	Context models features and systems exposing them	Limited [information]	16 ***2000-2006*** [context-aware systems]	An evaluation framework for context models
Ye et al. [45]	CM	Compare and contrast the most popular context ontologies	Limited [information]	5 ***2003-2004*** [ontologies]	Highlighting deficiencies of context ontologies
Kjaer [46]	C-aware [CM, CR]	Features of context-aware middleware systems (CAM)	Limited [overview]	9 ***2000-2004*** [middleware systems]	A taxonomy for categorizing properties of CAM
Najar et al. [47]	CM	Review of context models proposed in different domains	Limited [information]	8 ***2003-2008*** [context models]	A framework for analyzing context models
Perttunen et al. [48]	CM, CR	Context representation and reasoning in pervasive computing	Limited [overview]	36 ***1999-2008*** [projects]	Opportunities and limitations in current solutions

Saeed and Waheed [49]	C-aware [all stages]	Features of context-aware middleware architectures (CAM)	Limited [overview]	15 ***1998-2007*** [context-aware architectures]	Strengths and weaknesses of CAM
Vanathi and Uthariaraj [50]	CM	Features of ontology based context models in context-aware computing	Limited [information]	5 ***2003-2004*** [ontologies]	Ontologies evaluated according to design principles
Bauer [51]	CM	Comparison and validation of context meta-models	Appreciable [3 hierarchy levels]	13 ***1994-2010*** [context meta-models]	Similarities and differences of context models
Makris et al. [52]	C-aware [all stages]	Context-aware mobile and wireless networking (CAMoWiN)	Limited [overview]	±60 ***1997-2011*** [CAMoWiN research]	Components of a typical CAMoWiN architecture
Schuster et al. [53]	CM	Pervasive social computing applications	Limited [specific, social context]	43 ***2003-2011*** [systems]	A taxonomy to classify pervasive social context
Koç et al. [54]	CM	Research conducted on context modelling in a short period of time (2005-2013)	Limited [overview]	18 ***2005-2013*** [context models]	There's no methodology or a common language to model context
Our review	CM	Different context models specifically in context information, categorization schemes, formalisms and reusability	Extensive [Generic, different entities and context information]	138 ***2001-2015*** [context ontologies]	Consolidate past research covering its applicability from a service perspective

As can be seen in Table 4, none of the works is oriented to consolidate context knowledge generated for the last 14 years. Only Bauer [51] made an effort to provide a 3-level hierarchy of context information classes but: she considered a very limited amount of context models; semantic issues were not considered; reusability resources were not analyzed. Some remainder works focused on evaluating context models and their applications, while others were centered on context-awareness analyzing and evaluating how the stages of the context life cycle are managed. Consequently, we confirmed the need to consolidate existing research on context modelling based on ontologies from a service-oriented computing perspective.

2.2.2 Research questions of the survey

To formulate the research questions, we used the goal of the review [37] and the PICO (Population, Intervention, Comparison and Outcome) criteria [38]. Using PICO, research questions are defined by means of these criteria allowing dissecting useful keywords to structure the main search string in order to obtain more evidence from the study field.

The goal of this SM is to assess the context knowledge pieces defined in existing context models based on ontologies to identify their reusability in proposals from the perspective of service-oriented computing. In this context, the *comparison* criterion is out of the scope of this review since we are not interested on comparing context models. Therefore, the main research question was finally formulated as shown in Table 5. As it can be seen, the main research question was highly generic, so that we refined it into specific sub-questions considering interests and motivations of the review.

Table 5. Research questions of the systematic mapping

Main research question	
In the field of services (P), do ontology-based context models (I) proposed so far provide an adequate and structured set of context knowledge pieces to be reused in prospective proposals that can define their context (O)?	
Specific sub-questions	
Research sub-question	*Interest and motivation*
RQ1.1. What is the chronological overview of the research done so far in ontology-based context models?	Identify the proposals in the field, find their interrelationships and distribute them along time to find any significant trend, considering also their provenance (academy or industry)
RQ1.2. What are the characteristics of the proposed ontology-based context models?	Make explicit the main characteristics of these context models in terms of size, structure and completeness
RQ1.3. Which classes of context information and entities are the most addressed in ontology-based context models?	Identify aspects related to the scope of these context models, such as: what are the contexts and entities attracting more attention from researchers, since it may help to understand their priorities and eventually some research gaps
RQ1.4. What are the most consolidated classes of context information and entities in ontology-based context models?	We aim at identifying the most recurrent definitions of classes of context information and entities, which in some sense could be considered as the starting point of any new future proposal

2.2.3 Bibliographic sources

The search process conducted in this SM comprised automatic and manual searches. The automatic search was carried out by using several digital libraries (DLs), and the manual search through collecting the works from specific journals and conferences of the field of interest. Advantages and drawbacks of both approaches were analyzed through a case study by Kitchenham et al. [55]. Based on this analysis, we decided to integrate both strategies by accomplishing an automatic search in the selected DLs and complementing the results with manual searches to the most relevant conferences and journals in case that some conference edition or journal issue was missing in such DLs. To select the bibliographic sources relevant to this review, we followed the study and selection criteria provided by Dieste et al. [56]. As a result, the selected DLs were *Scopus*, *IEEE Xplore* and *ACM Digital Library*.

Concerning manual searches, we identified a list of journals and conferences relevant to this study. We targeted venues mainly in the field of software services and context modeling such as information systems and technologies, pervasive and ubiquitous computing, human-computer interaction, modeling and data engineering, and semantic web (see Table 6). These venues were selected from the top-ranked list based on the JCR impact factor for journals and the CORE-A status[4] for conferences. It is important to keep in mind that these sources were identified with the purpose of checking completeness of the study, but the main bibliographic corpus came from the automatic searches, due the broad spectrum of the selected DLs.

[4] There is no standard procedure to select the list of conferences. Although the CORE-A index, as any other, can raise controversy, we consider it as a good indicator for our purposes. See http://core.edu.au/index.php/conference-rankings.

Table 6. Journals and conferences classified by category

Categories	Journals	Conferences
Information technologies	Information Systems (IS); Information Sciences; Information Technology (JIT); Computer; IEEE Software; Advanced Engineering Informatics; European Journal of Information Systems (EJIS); Data & Knowledge Engineering (DKE); Information Science (JIS); Systems and Software (JSS); IEEE Transactions on Software Engineering (TSE); Journal of the ACM; Communications of the ACM.	International Conference on Information Systems (ICIS); Conference on Information and Knowledge Management (CIKM); Americas Conference on Information Systems (AMCIS); European Conference on Information Systems (ECIS); Conference on Advanced Information Systems Engineering (CAiSE); International Conference on Computational Science (ICCS); Enterprise Distributed Object Computing Conference (EDOC).
Pervasive and ubiquitous computing & Adaptive systems	IEEE Pervasive Computing; Pervasive and Mobile Computing (PerCom); Personal and Ubiquitous Computing (PUC); Adaptive Behaviour; Ambient Intelligence and Smart Environments (JAISE); ACM Transactions on Autonomous and Adaptive Systems (TAAS).	Pervasive Computing and Communications (PERCOM); Pervasive Computing (PERVASIVE); Ubiquitous Computing (UbiComp); Mobile and Ubiquitous Systems: Networks and Services (Mobiquitous).
Services	IEEE Transactions on Services Computing (TSC); International Journal of Web and Grid Services (IJWGS).	Conference on Services Computing (SCC); IEEE International Conference on Web Services (ICWS); International Conference on Service Oriented Computing (ICSOC).
Human computer interaction	International Journal of Human-Computer Studies; ACM Transactions on Computer-Human Interaction.	Human-Computer Interaction (HCI); IFIP TC13 Conference on Human-Computer Interaction (Interact).
Modelling	Applied Ontology.	Conceptual Modelling (ER); Formal Ontology in Information Systems (FOIS).
Web and semantic web	ACM Transactions on the Web (TWEB); Journal of Web Semantics (JWS); World Wide Web-Internet and Web Information Systems (WWW).	World Wide Web (WWW); Web Information Systems Engineering (WISE); International Semantic Web Conference (ISWC); Extended Semantic Web Conference (ESWC).
Surveys	ACM Computing Surveys (CSUR); IEEE Communications Surveys and Tutorials.	

Conferences and journals shown in Table 6 were reviewed to verify that all their editions from 2001 to 2015 were indexed and published in the selected DLs. The period from 2001 to 2015 was selected as starting point since it is possible to identify a consolidation during this period of current research topics on context and services such as Internet of Things, *-as-service and Service Oriented Computing. This verification allowed us to identify when to apply automatic or manual searches, i.e. when the journal or conference was published with all its editions in some of the selected databases means that it is sufficient to apply an automatic search, otherwise, it is necessary to apply manual searches for each missing edition, leading to searches in a specific DL of a conference or journal.

The results obtained are depicted in Table 7, we omitted conferences and journals that were indexed and published in all of their editions such as Information Science, IEEE Pervasive Computing, Adaptive behavior, and so on. Hence, it was only specified those conferences and journals involving manual searches such as TSC, IJWGS, WWW, etc., having editions not included in the databases. This issue was found in a higher proportion in conferences than journals: 7 journals and 17 conferences required manual searches because at least one edition was missing in all the DLs used.

Table 7. Manual and automatic searches

Journal	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
IEEE Commun. Surv. Tutor.	NP	M	M	M	A	A	A	A	A	A	A	A	A	A	A
IEEE Trans. Serv. C.	NP	NP	NP	NP	NP	NP	NP	M	A	A	A	A	A	A	A
Int. J. Web Grid Serv.	NP	NP	NP	NP	M	M	A	A	A	A	A	A	A	A	A
World Wide Web	M	M	A	A	A	A	A	A	A	A	A	A	A	A	A
ACM Trans. Comput.-Hum. Int.	M	M	A	A	A	A	A	A	A	A	A	A	A	A	A
Pers. Ubiquitous C.	M	M	M	M	A	A	A	A	A	A	A	A	A	A	A
Applied Ontology	NP	NP	NP	NP	M	M	M	M	M	M	A	A	A	A	A
Conference	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
ICIS	M	M	M	M	A	A	A	A	A	A	A	A	A	A	A
PERVASIVE	NP	M	NP	M	A	A	A	A	A	A	A	A	NY	NY	NY
UbiComp	M	M	M	M	M	A	A	A	A	A	A	A	A	A	A
AMCIS	NP	NP	M	M	A	A	A	A	A	A	A	A	A	A	A
HCI	NP	NP	NP	M	NP	NP	M	M	M	M	M	M	M	A	M
ECIS	M	M	M	M	A	A	A	A	A	A	A	M	M	A	M
ICWS	NP	NP	M	A	A	A	A	A	A	A	A	A	A	A	A
Interact	NP	NP	M	NP	M	NP	M	NP	M	NP	M	NP	M	M	M
CAiSE	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ICCS	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ER	M	M	M	M	M	M	M	M	M	M	A	M	M	M	M
FOIS	A	NP	NP	NP	NP	M	NP	M	NP	M	NP	M	NP	M	NY
MobiQuitous	NP	NP	NP	A	A	A	A	M	A	M	M	M	M	A	A
ICSOC	NP	NP	M	A	M	M	M	M	M	M	M	M	M	M	M
WISE	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ISWC	M	M	M	M	M	M	M	M	M	M	A	M	M	M	M
ESWC	NP	NP	NP	M	M	M	M	M	M	M	M	M	M	M	M

Legend: A - Automatic, M - Manual, NP - Unpublished, NY - Not Yet Available

2.2.4 Keywords used

To compose the main search string, we acquired the keywords from the PICO criteria specified in Section 2.2.2. Specifically, *Population* and *Intervention* criteria are used to extract them. In the context of this study, *Comparison* was discarded in the establishment of the research questions (Section 2.2.2) and *Outcome* was not considered because it is not based on a particular measurement in the research questions. As stated by Petersen et al. [37][40] and Kitchenham et al. [57], the *Outcome* criterion is not always applicable. Hence, from each term of the *Population* and *Intervention* criteria, we identified the keywords used to build the search string as depicted in Table 8.

Table 8. Keywords and search string

Criteria	Keyword	Variants
Population	Service	"service", "services"
Intervention	Context model	"context model", "context models", "contexts model", "contexts models"; "context ontology", "context ontologies", "contexts ontology", "contexts ontologies"; "context taxonomy", "context taxonomies", "contexts taxonomy", "contexts taxonomies"; "context hierarchy", "context hierarchies", "contexts hierarchy", "contexts hierarchies"
Search string		
("service" OR "services") AND ("context model" OR "context models" OR "contexts model" OR "contexts models" OR "context ontology" OR "context ontologies" OR "contexts ontology" OR "contexts ontologies" OR "context taxonomy" OR "context taxonomies" OR "contexts taxonomy" OR "contexts taxonomies" OR "context hierarchy" OR "context hierarchies" OR "contexts hierarchy" OR "contexts hierarchies")		

Note that to build the search string, variants inside *Population* and *Intervention* are interconnected through OR connectors (e.g. "service" OR "services") and finally, these criteria are joined through an AND connector. Since we are interested in ontology-based approaches, we added "context ontology" to the search string. Furthermore, according to Fernández-López et al., there are different levels to express an ontology referring to taxonomies or hierarchies [58], therefore we added also these terms. Last, in order to increase the quality and to improve the searches, we configured and simplified singular and plural forms of the search string by using the guidelines provided in each of the databases selected due to known limitations (e.g., length of the query in some databases).

2.2.5 Selection criteria

Once the bibliographic sources and the search string were specified, the search was conducted by title, abstract and keywords both in works obtained automatically or manually. After retrieving the results, we followed several steps to filter the candidates:

- **Filter by title.** Step to quickly identify and remove noise from results. After this selection, documents whose scope was clearly unrelated to context models were removed.
- **Filter by abstract.** Step used to delete works that although being related to context models, did not present an ontology-based context model as a contribution of the paper.
- **Filter by fast reading of full paper.** Step to discard those papers that did not fulfil properly the following inclusion criteria: (1) defining explicitly the context model; (2) presenting a context model that can be applied from the perspective of the service-oriented computing, i.e., the model conceptualizes a process or some of the entities involved in the service provisioning and consumption (e.g. user or provider of a service, service composition, etc.).
- **Snowballing.** Step that gathers further works during the SM process. To conduct this step, we employed backward snowballing that identifies relevant works from the reference list of the articles. We have included those referenced works that fulfil the previous inclusion criteria.

The results obtained from the searches are described as follows (see Fig. 2): 787 papers automatically from *Scopus*, 293 from *IEEE Xplore* and 72 from *ACM DL* yielding a total of 1.152 papers from which 220 were removed as they were duplicated, resulting 932 papers found automatically. Then, 111 papers were added from manual searches in selected venues (as explained in Section 2.2.3), obtaining 1.043 papers from which 712 were deleted by title and abstract, resulting in 331 papers to filter by fast reading. We discarded 17 papers that were not available through our University resources and whose authors didn't provide a copy under our request. Finally, the resulting papers after filtering by full paper were 145 and after adding 19 papers by snowballing, we obtained 164 papers to include in the SM. Afterwards, when analysing them in detail, we found a set of equivalent proposals, i.e., proposals from the same authors with a similar contribution; in this case we selected the most complete proposal to be evaluated and reviewed all of them to identify relevant information not considered in the chosen representative papers. The list of papers along the selection process is available at [59].

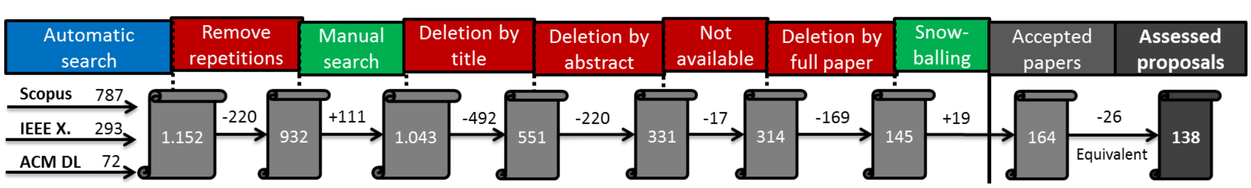


Fig. 2 Primary studies selection process

2.3 Results of the Review

In this section, the research questions specified in Section 2.2.2 are addressed. The results are an abstraction of the most important information retrieved by complete readings performed on each of the contributions selected. An extended report of the analyzed information (data extraction) from the proposals is provided at [59].

2.3.1 RQ1.1. What is the chronological overview of the research done so far in ontology-based context models?

2.3.1.1 Analysis per year

Fig. 3 displays a bar chart specifying papers discovered and selected in the review process by year of search comprising a period from 2001 to 2015. Since this analysis is focused on publications by year, in this research question we considered the 164 accepted papers instead of the 138 finally assessed, i.e., before discarding 26 equivalent papers (see Section 2.2.5). As depicted in Fig. 3, the amount of papers discovered by applying the search string resembles a bell curve with a peak in 2009. Even considering this decrease, the number of papers discovered in 2015 is still large enough as to consider the topic lively. At its turn, the distribution of the selected papers (which represent only 15% of the total amount of papers found in the search), presents a double bell curve with peaks in years 2007 and 2014.

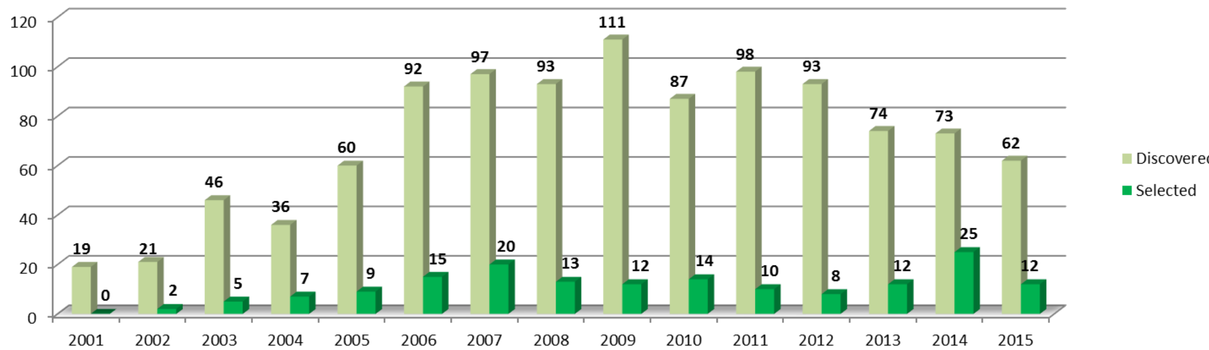


Fig. 3 List of papers discovered and selected by year period

2.3.1.2 Analysis per provenance

Fig. 4 shows a pie chart describing the percentages of contributions related to university, research centres and industry. As it can be seen, the provenance with largest contributions is linked to universities with 105 (76%) of the total amount of contributions in the field, which grows up to 121 (88%) if we consider collaborations. On the other side, proposals solely from industry are the exception: only 3 (2%), up to 7 (5%) if we include industry-university collaborations. It is worth noting that we identified this information based on the affiliation of the authors provided in the selected papers.

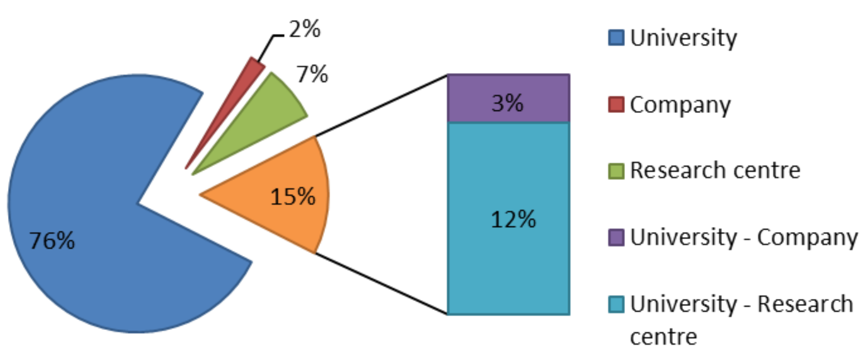


Fig. 4 Provenance distribution of selected papers

The geographic distribution of the selected papers over time is also another important discussion in this type of studies. To address it, we depict in Fig. 5 the contributions in context modeling distributed over the Earth's surface. As can be seen, during the period 2001 to 2015 these contributions have been spreading out in several geographic areas, i.e., different countries have attended the research field providing a variety of solutions. In this regard, one of the precursors in the period of time specified is Australia; however, at present it does not represent a country with large number of contributions in the field, in fact, only 5 of 164 are from there. This country in conjunction with countries with similar average of contributions such as USA, Canada, Spain, Italy, etc. represent 28% of the total contributions. Just above of India, Iran, Brazil, Portugal, etc. representing 14% of the solutions offered, being the countries with less participation in the field. In contrast to it, just China and Korea represent 34% of the total contributions with 21 and 17 respectively, presenting their first injection on context modeling in 2005. The remaining 24% of the selected papers is provided by UK, France, Germany and Malaysia as shown in the map.

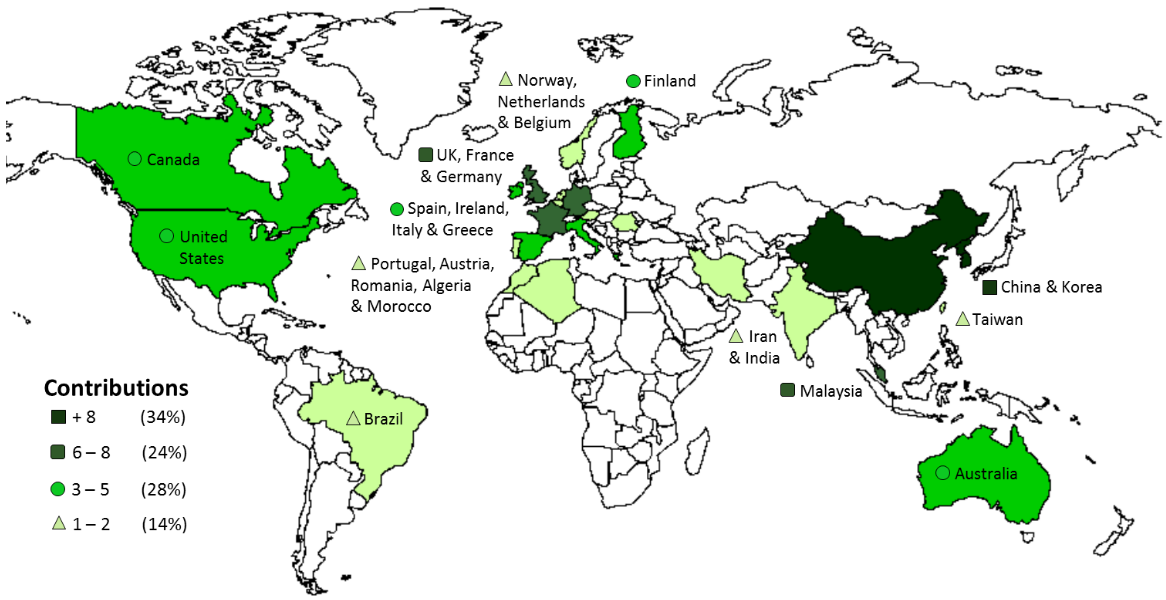


Fig. 5 Geographic distribution of selected papers

Combining both perspectives namely sector and geographic distributions, we also found interesting insights. Universities from China, France, Ireland, Italy, Greece, Brazil, Morocco, Iran, Romania, Belgium and Algeria were more influential in context modeling than companies. In fact, companies and research centers from these countries do not provided any contribution in the topic in the period of time from 2001 to 2015. Even collaborations between them were not identified. Contrary to this fact, countries with more collaborations between universities and other sectors (companies and research centers) were Korea, Malaysia, UK, Germany and USA. Similarly to them, but with less percentage of collaborations are Spain, Canada, Austria and Portugal. Countries with more contributions in the topic from companies and research centers with less presence of universities are Finland, Norway, Netherlands and India.

2.3.1.3 Evolutionary analysis

The goal of this section is to distribute the chronological evolution of the proposed context models for showing their relationships and identifying which ones are the most consolidated, the most influenced and the most influencing proposals. For this goal, we analyzed the chronological evolution of context models and depicted the research done in the field by means of a genealogical tree shown in Fig. 6. In the first row of the figure, we have identified different proposals out of the scope of this study, but which represent relevant background of the assessed proposals, regularly cited by them. The contribution by Maslow et al. [60] is the oldest one (1970), and although their proposal was not exactly a context model, it provided suitable vocabulary to be used in context modeling. Similarly, Uschold et al. [61] and the OpenCyc project [62] provided vocabulary from a specific ontology created for other purposes beyond context representation. Other authors with important presence in the field such as Schmidt et al. [63][64], Schilit et al. [65], etc. also exerted influence on the assessed proposals.

Regarding the period of time considered in this study, we found that the first two ontology-based context models proposed were issued in 2002 by Henricksen et al. [66] and Held et al. [67]. During this year, the context models proposed were individual contributions without influences by previous proposals in the field. However, 2003 witnessed the proposal of a first context model [68] influenced by previous contributions. Since then, some proposals have been developed based on previous models, reusing concepts, vocabulary and definitions. In consequence, most of the oldest context models have been updated or enhanced by other researchers. But also, isolated contributions have been identified with new proposals in the field such as the proposed by Ghannem et al. [69] and Mok and Min [70]. Other relevant findings have been classified as follows:

- *Standards*. During the review of context models, we sought for some standard context model used as starting point of other contributions. However, we did not find such a proper standard. The most similar contribution to a standard is the SOUPA ontology ("Standard Ontology for Ubiquitous and Pervasive Applications") [17]. This ontology is highly referenced in context research; in fact, as shown in Fig. 6, different proposals have been influenced by SOUPA, e.g. Cao et al. [12], Devaraju and Hoh [71], Pietschmann et al. [72], Cadenas et al. [73], Rubio et al. [74], Cabrera et al. [26]. We also noted that several proposals, included SOUPA, reuse a variety of vocabulary from standards developed for other purposes different than context, such as: DAML-Time [75], an ontology of temporal concepts; SUMO [76], an upper ontology that merges several ontologies from distinct domains providing hundreds of terms; OWL-S [77], semantic mark-up for Web services.
- *New approaches*. There are context models that have not considered previous contributions in their definitions nor have influenced other proposals. These proposals represent 82 of the 138 proposals assessed in the review (59% over the total), mostly concentrated in the period from 2005 to 2015 with the exception of 2012. The general purpose of these proposals is to provide new approaches or perspectives in context modeling not yet considered in previous contributions. However, although each model provides its own structure and formalism, we found context information classes with the same or similar meaning of another term. To avoid it, the proposals should have reused existing knowledge and provide the resources needed to facilitate future contributions.
- *Influences exerted by existing proposals*. Contributions that have influenced the definition of other context models represent 8% of the proposals assessed. As can be seen in Fig. 6, the context models with more significant impact in this sense are the oldest ones. Particularly, the most used context models are CONON [6] and SOUPA [17] which have been used to develop 9 and 6 context models respectively, followed by Hofer et al. [78], Preuveneers et al. [79] and Heckmann [80] influencing 2 context models each. The remaining contributions proposed by Henricksen et al. [66], Held et al. [67], Chen et al. [81], Xynogalas et al. [82], Bu et al. [83], Kim and Choi [84] influence 1 context model each. We also analyzed transitivity, namely if a context model X influences another context model Y and Y at its turn, influences another model Z, then we may say that X also influences Z. The chronology shows that this happened only in a small set of proposals assessed (7%) such as those influenced by Chen et al. [17], Preuveneers et al. [79] and Heckmann [80].
- *Proposals considering different contributions*. The context models in this category were developed from three perspectives, namely reusing vocabulary from an existing model with or without context purposes based on ontologies or on other formalisms, adopting the approach of an existing context model or a combination of them. In this regard, contributions considering more models for the development of their own proposals are provided by Chen et al. [17], Neto and Pimentel [85] and

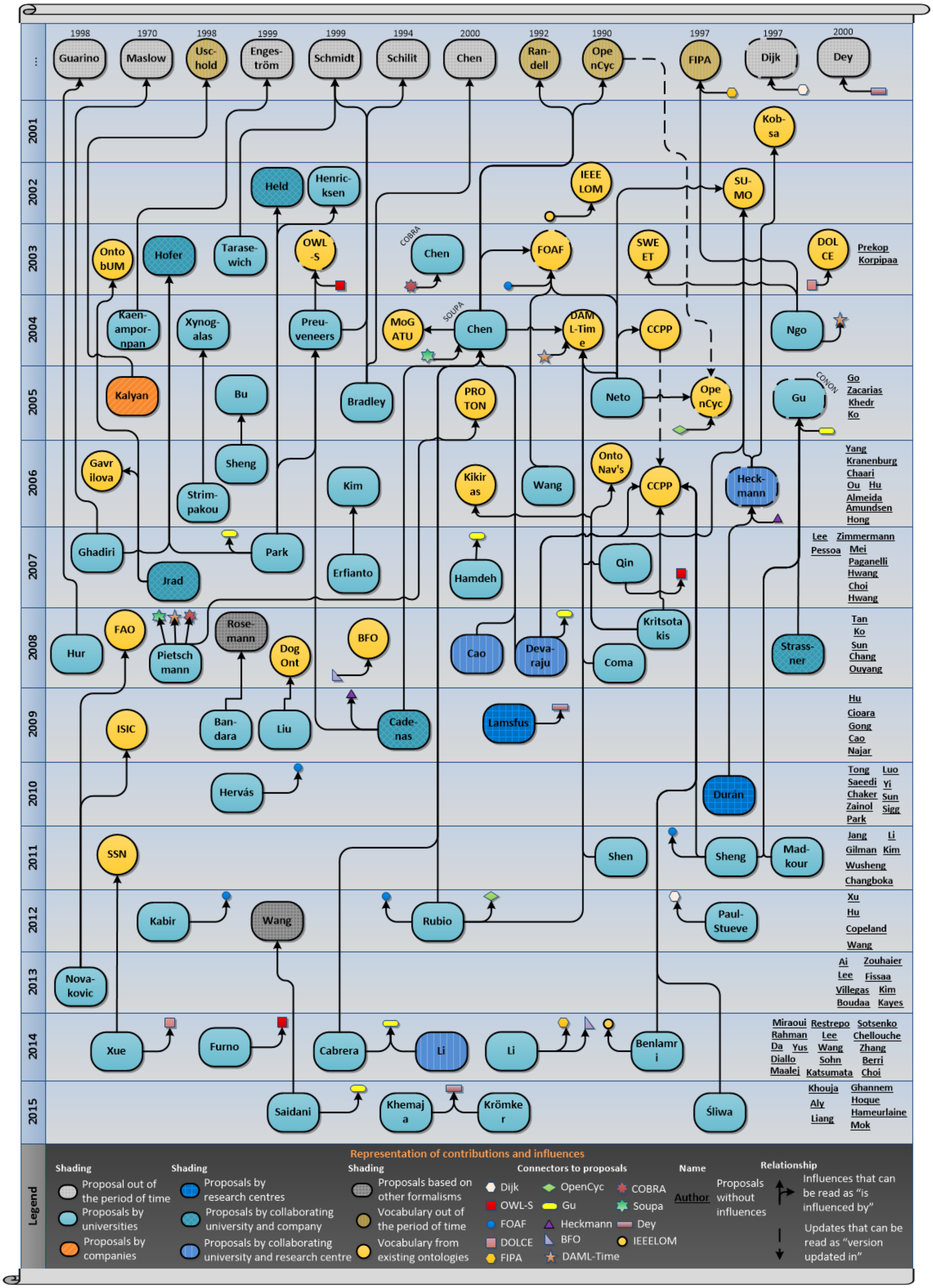


Fig. 6 Genealogical tree of ontology-based context models

Park and Kwon [86] with 5 considered proposals each. Meanwhile Chen et al. and Neto and Pimentel reuse vocabulary from existing ontologies not necessarily context oriented, such as FOAF [87], OpenCyc [62] and DAML-Time [75], Park and Kwon further adopts context model approaches from Henricksen et al. [66], Held et al. [67], Hofer et al. [78], Gu et al. [6], Preuveneers et al. [79]. Other proposals considering existing contributions in their models are Devaraju and Hoh [71], Pietschmann et al. [72] and Rubio et al. [74] with 4 proposals each, followed by Cadenas et al. [73], Preuveneers et al. [79], Ngo et al. [88], Bradley and Dunlop [89], Kritsotakis et al. [90] and Sheng et al. [91] with 3 proposals each.

Summing up, the chronological evolution of existing context models shows the need of a standard context model that can be considered as a consolidated basis of context knowledge for new proposals. This lack translates into different variation degrees among the models with respect to conceptualization, semantics in the primitives considered, structural formalizations and design patterns, among others. Although the existing variations can represent research solutions for different domain scenarios, the lack of such a common body of knowledge may represent additional costs (e.g., operation, engineering, maintenance, etc.) in the context-aware software development process. These costs will emerge when developers and modelers are faced to the construction of models from scratch that are not aligned with a common knowledge, postulating again models that cannot be reused or maintained for other developers.

2.3.2 RQ1.2. What are the characteristics of the proposed ontology-based context models?

An ontology-based context model provides structural characteristics and resources that make it suitable for certain purposes. Every proposal of context model should provide the required features to develop models easy to reuse in different situations. According to Fernández-López et al. [58], an ontology, to be reusable, must be well documented during the whole ontology development process and consider definitions of terms already specified whose semantic and implementation is coherent with the terms that are being defined. Therefore, to answer this research question, we analyzed and evaluated the structural characteristics and resources of the proposed ontology-based context models in terms of their size and definition coverage addressing completeness issues.

2.3.2.1 Size of structural characteristics and resources provided

According to the background literature on context modeling, a context model should consist of classes and properties, which altogether represent a central knowledge piece that should be defined and documented for increasing its reusability. Therefore, the size that we were interested to analyze and evaluate referred to the amount of context information that classes and properties (datatype and object properties) provide in the selected context models. Concerning object properties, we focused on those belonging to ad hoc binary relations (e.g. isLocatedAt, isAffectedBy, isUsedBy, etc.) since they represent relations between classes of an ontology (e.g. Person isAffectedBy Environment) and could support more powerful semantic analysis and reasoning [58][92]. Relations that belong to the development of basic concept taxonomies such as Subclass-of, Disjoint-Decomposition, Exhaustive-Decomposition, and Partition were not considered in this research question. Such relations that in some extent have to do with hierarchical structure of classes and subclasses are analyzed in Section 2.3.3.

2.3.2.1.1 Number of levels

The size of the context model was evaluated considering the amount of classes that provide context information and the number of levels of the class hierarchy (which we call depth level). The correlation between the number of classes and the depth level is depicted in Fig. 7. We highlight the following:

- From the total amount of classes (2,756) and depth levels (332) in 138 proposals, on average, the proposed context models have 19.97 classes and depth level of 2.4. The largest context models contain more than 40 classes and are arranged into 3 or 4 levels. The largest one by Ngo et al. presents up to 97 classes organized into 4 levels [88]. The smallest one by Lee and Kwon [93], Chaker et al. [94], and Henricksen et al. [66] present a context model of 3 classes in a single level.
- There are two main regions in the figure. At the bottom, left-most part, 61 models (44.2% of the total) are expressed in less than 15 classes distributed in at most 2 levels. At the top, right-most part, 26 models (18.8%) comprise at least 31 classes organized into 3 or 4 levels.
- Going a bit further in this direction, we observe a linear disposition of the results, which is steeper for few classes and gets flattened as the number of classes increases. Roughly speaking: models with [3, 10] classes tend to be organized in 1 level; models with [7, 18] and [11, 22], in 2 levels; models with [15, 26], in 3 levels; and models with more than 27 classes, in 4 levels. The only remarkable exception is the 7 proposals with more than 40 classes arranged in 3 levels.

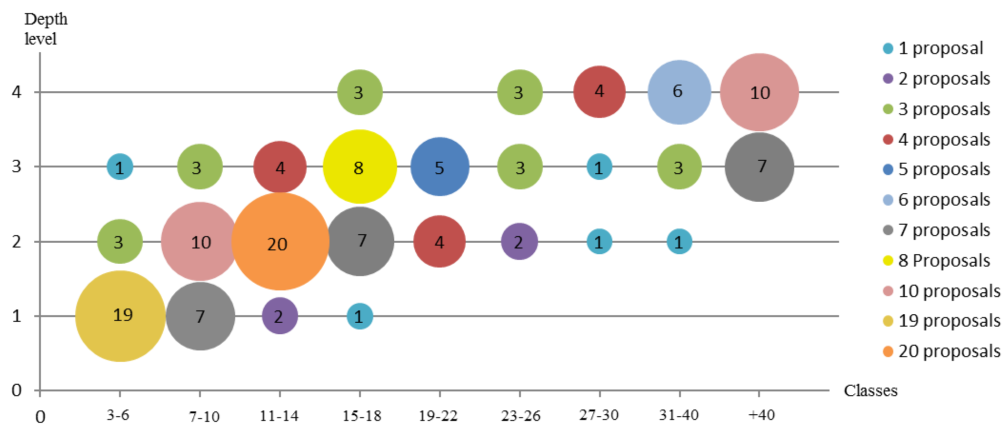


Fig. 7 Correlation map between context information classes and hierarchy levels

2.3.2.1.2 Number of datatype properties

Considering that the proposed context models should specify more details and resources to facilitate their reuse [95], we also evaluated the size of properties (datatype and object properties). Datatype properties link an individual to an XML Schema Datatype value or an RDF literal, i.e., they describe relationships between an individual and data values (e.g. the datatype property "age" links a person with his/her age that is a literal value). Contrary to them, object properties link an individual to an individual or a class to a class (e.g. the object property "hasParent" associates classes such as Person to a Person or in the case of individuals it associates an instantiation of the class Person). Hence, the evaluation was conducted by considering the amount of context information classes relative to the amount of datatype and object properties specified in the proposed context models. This was useful to correlate the quantity of properties provided in a certain number of context information classes.

Fig. 8 shows the size of context models regarding datatype properties. We can observe that most of the context models were developed with less than 10 datatype properties (109 proposals out of 138, i.e. 79.0%) and in particular, 83 of them (60.1% of the total) did not include any. In contrast to them, only 3 contributions (2.1%) were developed specifying more than 30 datatype properties being the context models proposed by Pietschmann et al. [72] and Hervás et al. [96] those providing larger number of datatype properties, 40 properties each.

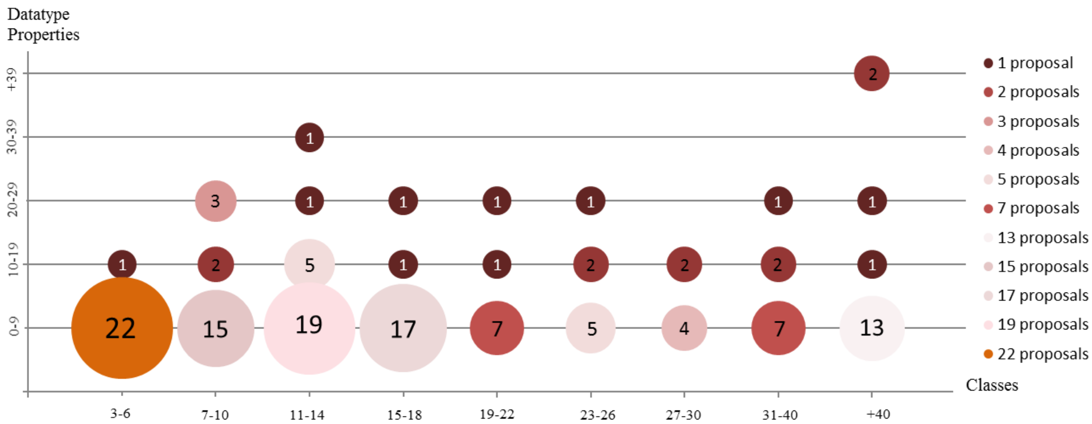


Fig. 8 Correlation map between context information classes and datatype properties

As it can be seen, the number of datatype properties is not correlated with the number of classes; in other words, the number of datatype properties does not depend on the number of classes. This was expected since each class of a model can be related to zero or many datatypes. Hence, it is possible to have a model with 5 classes and 5 datatype properties that relate only one of the classes with datatypes (e.g., in a model specifying 3 classes Time, Person and Activity only the Person class was related to different datatype properties such as hasName, hasAge, hasGender, etc. and it was not the case for Time and Activity).

2.3.2.1.3 Number of object properties

The size of object properties that associate classes to classes or individuals to individuals was evaluated by considering the amount of object properties, relative to the amount of classes in a model. The result is summarized in Fig. 9:

- The context models with more object properties are provided by Pietschmann et al. [72] and Hervás et al. [96] with 56 and 53 object properties linking 67 and 47 classes respectively. On the contrary, the context models with less object properties were identified in 66 proposals (47.8%) that do not provide any object property.
- Fig. 9 indicates that most of the context models were developed with a low range of object properties, less than 19 (114 proposals, 82.6%). From the remaining contributions, only 5 (3.6%) presented more than 30 object properties.
- Given the data retrieved, there is not a clear correlation between the number of object properties and classes, as also happened with datatype properties.

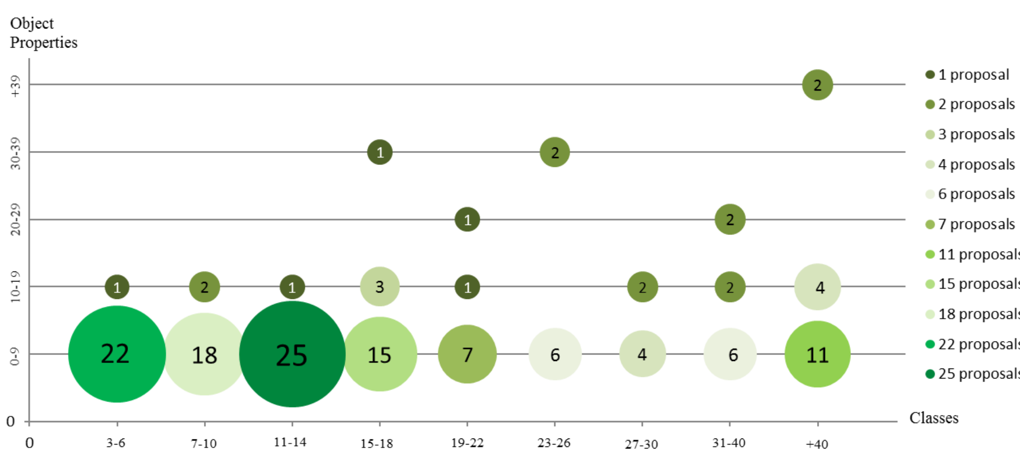


Fig. 9 Correlation map between context information classes and object properties

In fact, comparing the two correlation maps depicted in Fig. 8 and Fig. 9, we can observe that their shape is almost the same. We looked into the details of the proposals and we found a strong overlapping, even if it was not expected since the concepts in the two correlation maps do not depend on each other. In particular, 106 proposals (77%) were in the same area of both correlation maps, indicating that these approaches proposed context models that specified datatype properties almost in the same proportion as object properties. As a case worth to mention, from the 22 proposals that appear in the leftmost, bottom area corresponding to 0-9 properties and 3-6 classes, 21 of them are the same in both correlation maps indicating that proposals with more focused or smallest contributions to a context model, are balanced in terms of both types of properties. Similarly, the two proposals in the rightmost, top area are the same for both concepts.

2.3.2.2 Definition completeness of structural characteristics and resources provided

Definition completeness was evaluated on context information classes, datatype and object properties, in order to review the definition coverage of these characteristics, considering that each element in a model should be defined and associated with other resources of the model for better understanding in future reuses. Hence, the completeness evaluation of classes and properties was as follows: 1) for context classes, we evaluated the definitions and semantics provided for each class represented in the context model; 2) for datatype properties, we focused on identifying how many classes were or not related to literal values through datatype properties; 3) for object properties, we identified how many classes were or were not related to another class through object properties. The evaluation specified in 2 and 3 is because in the previous analysis we only identified the size of properties in the context models, but not the number of classes related to a property. It can be useful to identify if all the classes and properties specified in a context model have an active role for inferring or deducing context.

2.3.2.2.1 Completeness of context classes

The pie chart of Fig. 10 shows percentages of definition completeness regarding context information classes. As it indicates, only 20% of the presented proposals (28 contributions) have a unique and consistent definition for all the context information classes specified in the corresponding context model, i.e. 100% definition completeness, either by explicitly defining the context in the paper or by referencing to another bibliographic source that has the definitions.

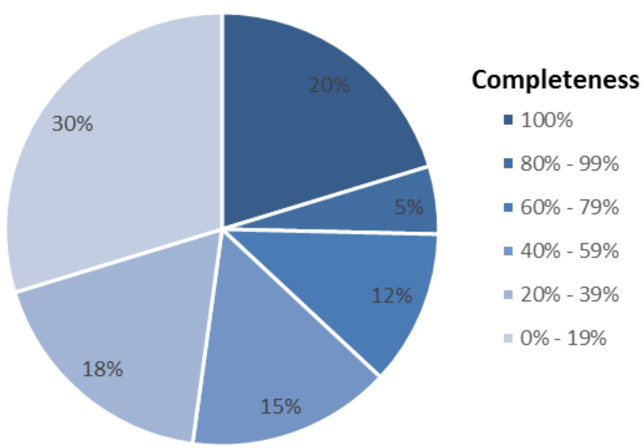


Fig. 10 Completeness of context information classes

Proposals with lower percentages of definition completeness present different issues, such as: some context information classes are not defined; some context models are based on other context models and although they are referenced, it is not specified which is the chosen definition for each context information class, leading to different definitions which are not consistent with each other; and the definitions on some context information classes are too vague or ambiguous. Note that as much as 30% of the proposals (41 contributions) have completeness level below 20%, which clearly is a very bad result. In fact, 20% of the proposals (27 contributions) do not provide any definition of context information classes.

2.3.2.2.2 Completeness of datatype and object properties

Definition completeness regarding datatype and object properties is illustrated in the pie charts of Fig. 11. The findings related to this completeness analysis indicate a percentage of properties that were defined in a model but were not associated with a class of the model (e.g., if a model specifies a size of 10 properties but only 6 are associated with a class, it means that the 40% of the properties lack of range, i.e. they are defined but not related to any class). Hence, reusing a context model that does not provide 100% of completeness indicates that a set of properties should be reanalyzed to understand its domain and range of application (e.g., FOAF provides several properties that are not associated nor maintained causing reuse problems).

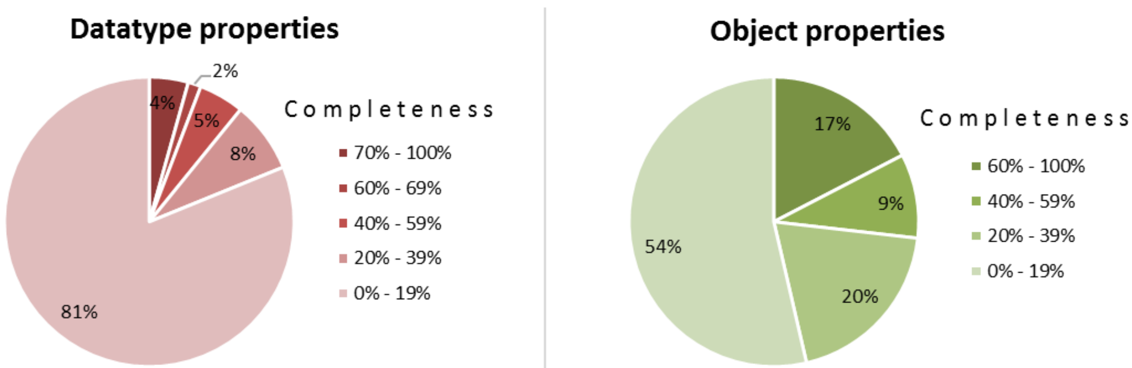


Fig. 11 Definition completeness of datatype and object properties

According to this criterion, the pie chart on the left in Fig. 11 indicates that only 4% of context models (6 proposals) have provided at least a datatype property to relate between 70% and 100% of their context information classes with datatypes. The rest has lower percentages and remarkably 81% of the approaches (112 contributions) do not reach a 20% level of completeness, pointing out a severe deficiency: most context models do not provide axioms involving datatype properties such as "each instance of the Location class must have an xsd:string data value for the *hasCoordinates* datatype property".

The pie chart on the right indicates that 17% of the context models (24 contributions) provide between 60% and 100% completeness of object properties to associate context information classes, representing a higher proportion than datatype properties but still low considering that the 73% of the context models (101 contributions) specify object properties below 40% of completeness. It indicates that a higher percentage of context models do not associate their classes with object properties to specify axioms such as "every instance of the Environment class must be related to instances of the Place class". However, the union between datatype and object properties indicates that a lower set (12 contributions, i.e. 9%) of context models relates 100% of their classes with a datatype or object property. An extended report of the data used for this analysis can be found at [59].

Table 9 condenses the most relevant information obtained throughout these two first research questions comprising the chronological overview and characteristics of the 138 reviewed proposals.

Table 9. Characteristics of the context models

Proposal	Formalism	Development method	Depth level	Clas-ses	Data-type p.	Object p.	Definition completeness		
							Context information	Datatype property	Object property
1. Śliwa and Gleba [2015]	Ontology-based	**149**	3	44	6	14	27%	14%	25%
2. Krömker and Wienken [2015]	Context taxonomy	**170**	1	8	0	0	50%	0%	0%
3. Aly et al. [2015]	Ontology-based	none	3	19	11	9	32%	21%	37%
4. Khemaja and Buendia [2015]	Ontology-based	**170**	3	23	0	7	43%	0%	39%
5. Hoque et al. [2015]	Ontology-based	none	4	51	5	8	0%	6%	12%
6. Hameurlaine et al. [2015]	Ontology-based	none	3	22	9	11	45%	18%	32%
7. Khouja and Juiz [2015]	Ontology-based	none	2	11	0	5	0%	0%	55%
8. Liang-Liang [2015]	Ontology-based	none	1	5	0	4	100%	0%	100%
9. Saidani et al. [2015]	Ontology-based	**118, 169**	3	17	20	4	41%	59%	30%
10. Ghannem et al. [2015]	Ontology-based	None	2	17	0	13	6%	0%	76%
11. Mok and Min [2015]	Ontology-based	None	1	4	0	0	0%	0%	0%
12. Restrepo et al. [2014]	Ontology-based	None	1	4	7	0	100%	100%	0%
13. Lee et al. [2014]	Context taxonomy	None	2	14	0	0	100%	0%	0%
14. Xue et al. [2014]	Ontology-based	**139, 140**	4	37	1	11	51%	3%	30%
15. Wang and Tang [2014]	Ontology-based	None	2	5	0	0	60%	0%	0%
16. Cabrera et al. [2014]	Ontology-based	**118, 127**	1	11	0	8	100%	0%	64%
17. Sohn et al. [2014]	Ontology-based	None	3	13	8	0	8%	62%	0%
18. Yus et al. [2014]	Ontology-based	None	3	9	2	4	0%	11%	44%
19. Choi et al. [2014]	Ontology-based	None	4	59	0	0	26%	0%	0%
20. Sotsenko et al. [2014]	Context taxonomy	None	2	17	0	0	24%	0%	0%
21. Li and Chen [2014]	Ontology-based	**141, 142**	2	8	0	0	88%	0%	0%
22. Chellouche et al. [2014]	Ontology-based	None	2	36	0	17	11%	0%	61%
23. Zhang et al. [2014]	Ontology-based	None	3	31	20	0	0%	52%	0%
24. Berri [2014]	Context taxonomy	None	2	10	0	0	100%	0%	0%
25. Li et al. [2014]	Ontology-based	**118**	2	9	0	0	33%	0%	0%
26. Katsumata [2014]	Ontology-based	None	2	12	3	0	33%	8%	0%
27. Da et al. [2014]	Ontology-based	None	1	4	0	4	50%	0%	100%
28. Benlamri and Zhang [2014]	Ontology-based	**149, 172**	2	26	14	31	65%	31%	81%
29. Miraoui [2014]	Ontology-based	none	2	14	0	9	29%	0%	57%
30. Rahman et al. [2014]	Context taxonomy	none	3	30	0	0	23%	0%	0%
31. Diallo et al. [2014]	Ontology-based	none	4	47	3	2	13%	0%	68%
32. Maalej et al. [2014]	Ontology-based	none	3	26	8	9	50%	4%	54%
33. Furno and Zimeo [2014]	Ontology-based	**155**	3	18	0	5	22%	0%	28%
34. Novakovic and Huemer [2013]	Ontology-based	**143, 144**	4	16	0	0	13%	0%	0%
35. Ai et al. [2013]	Ontology-based	None	3	18	0	8	61%	0%	50%
36. Lee and Kwon [2013]	Ontology-based	None	1	3	0	0	100%	0%	0%
37. Villegas and Müller [2013]	Ontology-based	None	4	25	24	37	100%	28%	72%
38. Kayes et al. [2013]	Ontology-based	None	2	16	0	7	31%	0%	63%
39. Boudaa et al. [2013]	Ontology-based	None	2	11	2	2	55%	18%	27%
40. Kim et al. [2013]	Ontology-based	None	2	21	0	0	0%	0%	0%
41. Fissaa et al. [2013]	Ontology-based	None	3	18	0	0	22%	0%	0%
42. Zouhaier et al. [2013]	Ontology-based	None	1	8	0	7	38%	0%	100%
43. Kabir et al. [2012]	Ontology-based	**145**	1	4	0	4	100%	0%	100%
44. Rubio et al. [2012]	Ontology-based	**127, 145, 146, 147**	3	9	20	13	67%	89%	33%
45. Paul.and Wachsmuth [2012]	Ontology-based	**148**	1	8	0	0	0%	0%	0%
46. Hu et al. [2012]	Ontology-based	None	1	5	0	5	0%	0%	100%
47. Copeland and Crespi [2012]	Context taxonomy	None	1	6	0	0	100%	0%	0%
48. Wang et al. [2012]	Ontology-based	None	3	15	0	0	0%	0%	0%
49. Xu and Dong [2012]	Ontology-based	none	2	24	0	0	0%	0%	0%
50. Shen and Cheng [2011]	Ontology-based	**146**	4	27	5	0	44%	7%	0%
51. Sheng et al. [2011]	Ontology-based	**118, 145, 149**	2	14	0	4	100%	0%	29%
52. Madkour and Maach [2011]	Ontology-based	**118**	3	31	0	3	19%	0%	13%
53. Jang and Choi [2011]	Ontology-based	None	4	39	15	20	0%	13%	46%
54. Gilman et al. [2011]	Ontology-based	None	2	15	0	0	47%	0%	0%
55. Wusheng et al. [2011]	Ontology-based	None	2	7	0	0	71%	0%	0%

Proposal	Formalism	Development method	Depth level	Clas-ses	Data-type p.	Object p.	Definition completeness: Context information	Datatype property	Object property
56. Li [2011]	Ontology-based	None	2	13	10	8	54%	23%	54%
57. Kim et al. [2011]	Ontology-based	None	1	4	0	0	100%	0%	0%
58. Changboka et al. [2011]	Ontology-based	None	4	16	7	1	19%	13%	13%
59. Hervás et al. [2010]	Ontology-based	**145**	4	47	40	53	23%	30%	85%
60. Durán et al. [2010]	Ontology-based	**109**	2	12	19	4	58%	58%	42%
61. Tong et al. [2010]	Ontology-based	None	4	16	0	19	0%	0%	100%
62. Luo et al. [2010]	Ontology-based	None	1	4	0	0	75%	0%	0%
63. Yi et al. [2010]	Ontology-based	None	1	8	29	0	0%	88%	0%
64. Sun et al. [2010]	Ontology-based	None	2	11	15	2	36%	45%	27%
65. Sigg et al. [2010]	Ontology-based	None	3	67	0	0	0%	0%	0%
66. Park et al. [2010]	Ontology-based	None	2	12	17	9	17%	33%	75%
67. Zainol and Nakata [2010]	Ontology-based	None	3	13	38	0	31%	62%	0%
68. Chaker et al. [2010]	Context taxonomy	None	1	3	0	0	100%	0%	0%
69. Saeedi et al.[2010]	Ontology-based	None	2	11	0	0	18%	0%	0%
70. Bandara et al. [2009]	Ontology-based	**173**	2	12	10	0	100%	25%	0%
71. Liu et al. [2009]	Ontology-based	**150**	4	23	10	6	78%	13%	43%
72. Cadenas et al. [2009]	Ontology-based	**109, 127, 130**	1	10	0	11	100%	0%	100%
73. Hu and Li et al. [2009]	Ontology-based	None	2	12	29	8	33%	50%	33%
74. Cioara et al. [2009]	Ontology-based	None	2	5	0	0	60%	0%	0%
75. Gong et al. [2009]	Ontology-based	None	4	27	0	5	26%	0%	33%
76. Cao et al. [2009]	Ontology-based	None	2	9	14	4	0%	56%	44%
77. Najar et al. [2009]	Ontology-based	None	3	19	0	1	47%	0%	11%
78. Lamsfus et al. [2009]	Ontology-based	**170**	1	11	0	0	82%	0%	0%
79. Cao et al. [2008]	Ontology-based	**127**	2	7	0	1	57%	0%	29%
80. Hur et al. [2008]	Ontology-based	**151**	1	4	0	9	0%	0%	100%
81. Devaraju and Hoh [2008]	Ontology-based	**118, 127, 152, 149**	4	39	0	1	77%	0%	5%
82. Coma et al. [2008]	Ontology-based	**146**	3	14	2	2	86%	14%	14%
83. Strassner et al. [2008]	Ontology-based	**118**	2	19	0	0	53%	0%	0%
84. Tan et al. [2008]	Context taxonomy	**153**	4	46	0	0	13%	0%	0%
85. Sun et al. [2008]	Ontology-based	None	4	32	5	8	0%	9%	22%
86. Ouyang et al. [2008]	Ontology-based	None	2	18	6	5	61%	11%	28%
87. Kritsotakis et al. [2008]	Ontology-based	**149, 167, 168**	4	68	0	5	51%	0%	10%
88. Ko and Sim [2008]	Ontology-based	None	3	19	7	5	21%	11%	21%
89. Pietschmann et al. [2008]	Ontology-based	**127, 136, 146, 171**	4	67	40	56	100%	15%	63%
90. Chang et al. [2008]	Ontology-based	none	2	7	0	4	100%	0%	29%
91. Ghadiri et al. [2007]	Ontology-based	**132, 154**	4	39	0	0	28%	0%	0%
92. Park and Kwon [2007]	Ontology-based	**118, 130, 132, 137, 138**	1	15	0	0	80%	0%	0%
93. Erfianto et al. [2007]	Ontology-based	**107**	3	13	3	0	31%	15%	0%
94. Qin et al. [2007]	Ontology-based	**146, 155**	3	17	17	13	24%	29%	41%
95. Lee and Meier [2007]	Ontology-based	None	1	4	11	0	75%	100%	0%
96. Mei et al. [2007]	Context taxonomy	None	1	5	0	0	100%	0%	0%
97. Paganelli et al. [2007]	Ontology-based	None	2	11	0	6	55%	0%	55%
98. Zimmermann et al. [2007]	Context taxonomy	None	2	12	0	0	100%	0%	0%
99. Hwang et al. [2007]	Ontology-based	None	3	73	0	0	11%	0%	0%
100. Choi and Yoon [2007]	Context taxonomy	None	3	17	0	0	82%	0%	0%
101. Hwang et al. [2007]	Ontology-based	None	1	6	0	7	17%	0%	100%
102. Jrad et al. [2007]	Ontology-based	**165, 166**	2	14	0	0	93%	0%	0%
103. Pessoa et al. [2007]	Ontology-based	None	3	16	2	6	0%	6%	31%
104. Hamdeh and Ma [2007]	Ontology-based	**118**	4	49	2	5	6%	4%	12%
105. Strimpakou et al. [2006]	Ontology-based	**129**	2	18	0	32	67%	0%	78%
106. Sheng et al. [2006]	Ontology-based	**120**	2	9	7	5	100%	11%	67%
107. Kim and Choi [2006]	Ontology-based	None	4	27	15	15	0%	22%	37%
108. Wang et al. [2006]	Ontology-based	**145**	1	7	25	0	100%	57%	0%
109. Heckmann [2006]	Ontology-based	**152, 156**	3	39	0	0	0%	0%	0%
110. Yang et al. [2006]	Ontology-based	None	4	49	0	0	29%	0%	0%
111. Kranenburg et al. [2006]	Context taxonomy	None	1	6	0	0	0%	0%	0%
112. Chaari et al. [2006]	Ontology-based	None	1	7	0	0	0%	0%	0%
113. Ou et al. [2006]	Ontology-based	None	3	42	0	0	7%	0%	0%
114. Almeida et al. [2006]	Ontology-based	None	2	11	0	10	64%	0%	100%

Proposal	Formalism	Development method	Depth level	Classes	Datatype p.	Object p.	Definition completeness		
							Context information	Datatype property	Object property
115. Hu and Moore [2006]	Ontology-based	None	2	11	0	0	100%	0%	0%
116. Amundsen and Eliass. [2006]	Context taxonomy	None	2	10	17	0	100%	70%	0%
117. Hong et al. [2006]	Ontology-based	None	3	10	0	0	0%	0%	0%
118. Gu et al. [2005]	Ontology-based	None	2	13	0	6	0%	0%	31%
119. Kalyan et al. [2005]	Ontology-based	**157**	3	20	0	0	40%	0%	0%
120. Bu et al. [2005]	Ontology-based	None	4	26	1	6	0%	4%	27%
121. Bradley and Dunlop [2005]	Context taxonomy	**158, 159, 160**	2	16	0	0	75%	0%	0%
122. Neto and Pimentel [2005]	Ontology-based	**145, 146, 147, 149, 152**	3	26	4	1	85%	8%	8%
123. Go and Sohn [2005]	Ontology-based	None	1	5	0	0	100%	0%	0%
124. Zacarias et al. [2005]	Context taxonomy	None	1	4	0	0	100%	0%	0%
125. Khedr and Karmouch [2005]	Ontology-based	None	4	30	18	15	23%	20%	30%
126. Ko et al. [2005]	Ontology-based	None	3	41	20	10	32%	4%	24%
127. Chen et al. [2004]	Ontology-based	**145, 146, 147, 161, 162**	2	20	26	24	100%	35%	75%
128. Kaenampor. and Neill [2004]	Ontology-based	**163**	2	22	0	0	73%	0%	0%
129. Xynogalas et al. [2004]	Context taxonomy	None	2	28	0	0	100%	0%	0%
130. Preuveneers et al. [2004]	Ontology-based	**155, 158, 159**	4	38	15	24	45%	16%	55%
131. Ngo et al. [2004]	Ontology-based	**141, 146, 164**	4	97	5	19	30%	1%	25%
132. Hofer et al. [2003]	Context taxonomy	None	1	5	0	0	100%	0%	0%
133. Tarasewich [2003]	Context taxonomy	**158**	3	54	7	0	78%	2%	0%
134. Prekop et al. [2003]	Ontology-based	None	2	7	0	0	43%	0%	29%
135. Korpipaa et al. [2003]	Ontology-based	None	2	12	0	0	0%	0%	0%
136. Chen et al. [2003]	Ontology-based	None	3	41	12	13	59%	7%	22%
137. Henricksen et al. [2002]	Ontology-based	None	1	3	6	14	100%	100%	100%
138. Held et al. [2002]	Ontology-based	None	2	5	0	0	0%	0%	0%

Identifiers: **139.** SSN (Barnaghi et al. 2011); **140.** DOLCE (Gangemi et al. 2003); **141.** FIPA 2000; **142.** BFO (Born et al. 2008); **143.** FAO (Caracciolo et al. 2008); **144.** ISIC 2009; **145.** FOAF (Brickley and Miller 2003); **146.** OWL-Time/DAML-Time (Pan and Hobbs 2004); **147.** OpenCyc (Lenat and Guha 1990); **148.** Dijk 1997; **149.** CCPP (Klyne et al. 2004); **150.** DogOnt (Bonino and Corno 2008); **151.** Guarino 1998; **152.** SUMO (Pease and Niles 2002); **153.** From 78 papers that were not referenced; **154.** Maslow 1970; **155.** OWL-S 2003; **156.** Kobsa 2001; **157.** Uschold et al. 1998; **158.** Schmidt et al. 1999; **159.** Schilit et al. 1994; **160.** Chen and Kotz 2000; **161.** Randell et al. 1992; **162.** MoGATU (Perich 2004); **163.** Engeström et al. 1999; **164.** NASA Jet Propulsion Lab space ontology (http://sweet.jpl.nasa.gov/ontology/); **165.** OntobUM (Razmerita et al. 2003); **166.** Gavrilova et al. 2006; **167.** OntoNav's (Tsetsos et al. 2006); **168.** Indoor Navigation Ontology (Kikiras et al. 2006); **169.** Wang and Jiang 2012; **170.** Dey et al. 2000; **171.** PROTON (http://proton.semanticweb.org/); **172.** IEEELOM standard (http://ltsc.ieee.org/wg12/); **173.** Rosemann et al. 2008

2.3.3 RQ1.3. Which classes of context information and entities are the most addressed in ontology-based context models?

Throughout this question, we intended to analyze the most addressed context information classes and entities of the reviewed ontology-based context models. To address it, we grouped classes of context information and entities into hierarchies and synonyms that allowed us to compact the total of 2.756 terms coming from the 138 selected context models into 250 representative terms. This process involved the analysis of context term definitions provided in the proposals. From this analysis, we identified three representative structures as a pattern presented in the reviewed context models (see Fig. 12). These structures are described as follows:

(A) **Entity-dependent**. Entities in a context model precede context information classes, i.e. each entity specified is characterized by different types of context information classes (e.g. *User* is an entity that can be characterized by *State*, *Profile*, *Environment*, etc.). Therefore, though the definition of an entity is intended to be generic, definitions of context information classes depend on the entity class from which they are related. A total of 44 (31.88%) of the contributions follow this structure.

(B) **Entity-independent**. Entities and context information classes in a context model are also clearly separated, but to link them, relations such as aggregation or composition are necessary (e.g. entities such as *Person*, *Organization* and *Service* are related to context information classes such as *Profile*, *Activity* and *Task*). In this case, definitions of context information and entities classes are more generic. Only 16 (11.59%) of the selected proposals follow this structure.

(C) **Context-sensitive**. Context models do not clearly separate entities from context information classes or vice versa. Therefore, definitions of classes of context information and entities can be as follows: the term used to define an entity also describes different types of context information classes (e.g. the entity *Person* refers to all humans and describes their *Profiles*, *Status*, *Social relationships*, etc.); terms used to specify entities are used to describe a specific context information class (e.g. the entity *Person* describes the *Profile* of a user); a term of a context information class can be used to characterize different entities (e.g. *Profile* refers to the profile of a *Device*, *Person* and *Activity*). This is the most adopted pattern with 78 (56.52%) of the selected proposals following this structure.

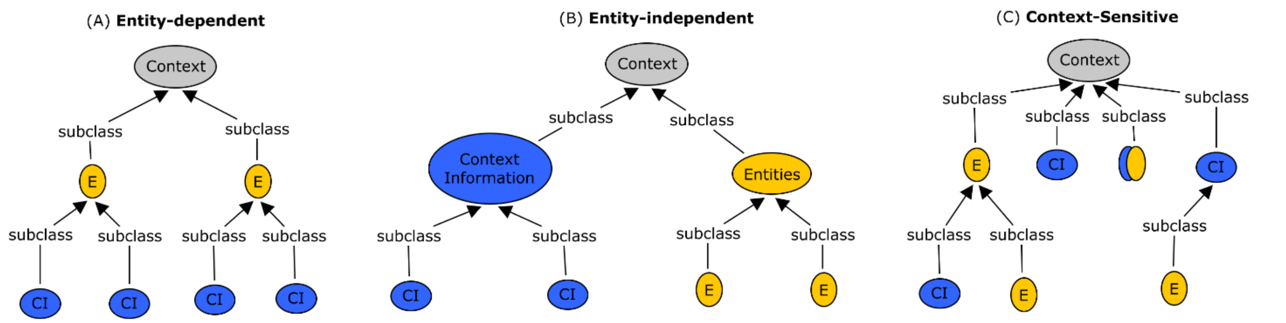


Fig. 12 Structure pattern to specify and describe entities and context information classes

The findings previously identified were useful to establish the following criteria for grouping synonyms and hierarchize classes of context information and entities in order to consolidate the 2.756 existing terms:

- Terms whose definition describes a generic entity class (e.g. "a *Service* is a resource to deliver value") or generic context information class (e.g. "*Activity* describes the activities of agents") are positioned in the same level, preceding specific classes of context information or entities (e.g. *Activity* precedes *UserActivity*).
- Terms defining solely entities (e.g. "a *Person* refers to all users of a system") are grouped as synonyms with terms defining entities that also describe context (e.g. "*Person* describes the context information class of a user") if and when they refer to a generic description and an exact or similar meaning.
- Terms whose definition describes an entity class that refers to a specific context information class (e.g., the term "*Person*" describes the profile of users) are grouped as synonyms with the term describing this specific context information class (e.g. "*Profile* describes the profile of a *Person*") if and when the entities characterized belong to the same type (e.g. *Person* and *User* belong to humans).
- Whether some definition was not provided, we consider the parent and children classes specified for each term, i.e. if the term "*Profile*" is not defined, but the parent class is a *Person* or its sub-classes are context information classes of a *Person*, clearly it refers to a specific context of a *Person*.

Table 10 shows the most representative synonyms and hierarchies as a result of applying the previous criteria in the selected context models, i.e. terms with more matches found in these models (see Table 11). In both tables we attempt to separate entities from context information classes in order to avoid blending them, issue presented in different proposals that might cause reusability problems. The complete analysis and an extended report of synonyms and hierarchies can be found at [59].

Table 10. Table of synonyms and hierarchies

Class	Second level	Third level	Synonyms
		Entities classes	
Agent			Actors, SocialEntity, User, UserDomain, Participant, Entity
	Person		People, OtherPeople, Individual, Personal, Human, HumanEntity, Visitor
		User	Who, User "fingerprints", UserViewInformation, Participants, Users, IntrinsicContext, UserDriver, UserInformation, UserEntity
	Organization		Org, OrganizationRelationship, Organizational
Resource			-
	Computational Entity		CompEntity, Computational, Computing, Computing&Connectivity, SystemViewInformation, ComputingDomain, Platform, ComputingEnvironment
		Device	Entity, ComputingEntity, Resources, ComputingDevices, Platform, Devices, how, Computation, ServiceObject, ComputationalDevice, TecnologicContext, DeviceContext
		Service	ServiceViewInformation, Services
		Network	Connectivity, NetworkEntity, NetworkConnectivity
		Context information classes	
Activity			Activities, what, UserActivity, ActivityProcess, SocialActivity, CommunityActivities, ActivityEntity
	Task		Tasks, CommunityTasks, AssociatedTasks
	Event		Events, ActivityEvent
	Agenda		ScheduledEvent, Calendar, Timetable, Scheduled, ScheduledActivity
Time			Time-related, When, Time, when, Temporal, TimeExpression, History, Time-based, TemporalInformation, TemporalInfo, TemporalThing, TimeContext
Environment			Environmental, ImmediateEnvironment, Extrinsic
	Physical Environment		EnvironmentalConditions, EnvironmentalSensorData, Environment, Climate, Physical, PhysicalAspects, Weather, Physics, PhysicalConditions, ClimateCondition, PhysicalContext, UserEnvironmentDescription, EnvironmentalContext, PhysicalEntity
	Social Environment		User'sSocialEnvironment, Social, SocialContext, HumanFactors
		Relation-ships	SocialRelations, SocialRelationships, SocialRelationshipData
Location			Co-location, Place, Where, LocationSensor, Space, Locality, LocationFeature, GeographicLocations, PhysicalLocation, Spatial_Context, LocationEntity, SpationTemporalContext
	Outdoor		OutdoorSpace, OutsideHome, OutdoorPlace
	Indoor		IndoorSpace, In-doorSpace, IndoorPlace
Profile			Profiles, DefaultProfiles, ProfileInfo
	UserProfile		Person, PersonProfile, PersonalInformation, ProfileInformation, UserPersonalDescription, UserCharacteristics, UserObject, PersonalProfile
	Preferences		UserPersonalPreferencesAndInterests, UserPreferences, UserPreferenceDescription
Role			PersonRole, DivisionOfLabour, AgentRole, SocialRole
StatesAnd			UserCurrentState, State, CurrentState, Status, StatusInfo,

Status			CurrentStateDescription, CurrentStatus
	Physiological States		Physiological, Physiologic, BioCondition, BiomedicalSensorData
	Emotion		UserEmotionalState, EmotionalState, Emotional, Emotional/Mental, EmotionalSituation, Mental state

As can be seen in Table 10, the context information class with higher number of synonyms is *Physical Environment* that considers 14 terms with exact or similar meaning, followed by *Location*, *Time* and *Device* with 12 terms each, *User* with 9 terms, and *Computational entity* and *UserProfile* with 8 terms each. Contrary to them, the class *Resource* is used consistently in all the selected context models, i.e. the term used to define it is always the same. We also identified three features regarding terms used to specify classes of context information and entities, described as follows: some terms maintain a regular variation of the word (e.g. "*Physiological states*" and most of its synonyms present a slight variation among them); some terms are highly generic or very specific (e.g. synonyms of "*Physical environment*" such as "*Environment*" and "*User environment description*"); finally, although the terms maintain the same meaning some of them seem to be very distinct (e.g. "*Environment*" and "*Extrinsic*").

These inconsistencies affect the reuse of context models especially if it is preferred to create new terms instead of verifying which models provide the vocabulary semantically coherent with the terms identified. In this sense, the integration of more than one model should be documented for every term whose definition is going to be used [58]. It is worth to remark that we identified terms representing entities; however, most of these terms were also used to describe context information classes in several proposals as noted above.

Regarding matches and hierarchy levels of classes of context information and entities depicted in Table 11, we found also inconsistencies related to how this information is hierarchized among the analyzed context models. As it can be seen in the table, none of the terms remains constant into the same hierarchy level; they appear at different levels at least once. Terms with slight variability into the specified hierarchy levels and that seem to be more consolidated in a single one, i.e. terms that in a higher percentage remain constant into the same hierarchy level, are *User*, *Organization*, *Tasks*, *Agenda*, *Environment*, *Outdoor*, *Indoor* and *UserProfile*. Note that we have hierarchized classes of context information and entities based on the consolidated hierarchy levels of the terms and considering generic definitions that can include a higher number of terms. In this sense, although the term *User* provides 34 matches at first level hierarchy, *Person* and *Agent* are more generic according to their definitions. A special case was identified with the term *Profile*: only 6 models (4.3%) provides a generic definition of this context information class, most of proposals refer to a *User profile*. However, we consider the term *Profile* in the hierarchy for grouping more terms such as *Device profile*, *Activity profile*, etc.

Table 11. Table of matches and hierarchies

Class	Second level	Third level	Matches	Hierarchy levels			
				1st	2nd	3rd	4th
Entities classes							
Agent			22	14	5	2	1
	Person		36	15	14	5	2
		User	41	34	6	1	0
	Organization		9	1	7	1	0
Resource			12	5	5	2	0
	ComputationalEntity		12	5	5	2	0
		Device	49	22	19	6	2
		Service	28	14	9	4	1

		Network	30	14	9	5	2
Context information classes							
Activity			68	34	27	5	2
	Task		19	5	12	0	2
	Event		14	6	6	2	0
	Agenda		19	3	12	4	0
Time			71	34	31	5	1
Environment			24	18	5	1	0
	PhysicalEnvironment		32	14	15	3	0
	SocialEnvironment		15	7	6	2	0
		Relationships	11	3	5	2	1
Location			92	40	40	11	1
	Outdoor		14	0	5	8	1
	Indoor		14	0	5	8	1
Profile			6	5	1	0	0
	UserProfile		43	12	22	7	2
	Preferences		25	8	9	7	1
Role			25	6	11	6	2
StatesAndStatus			12	4	4	4	0
	PhysiologicalStates		10	1	5	4	0
	Emotion		12	0	7	5	0

Once classes of context information and entities were grouped and hierarchized, we analysed which is the most addressed context information class of the selected context models. To do so, we used the first level classes previously established as a reference for the comparison. To evaluate the coverage of these classes on each context model, we defined the following criteria:

- *Explicitly defined without divisions* (✓). Describes whether a class is explicitly defined in the context model.
- *Explicitly defined with divisions* (✓*). Describes whether a class is explicitly defined in the context model with further subclasses.
- *Partially defined "reduced"* (P). Describes whether a class is not explicitly defined, but the model has a class or subclass which can be classified into this class.
- *Partially defined "extended"* (P*). Describes whether a class is not explicitly defined, but the context model has different classes or subclasses which can be classified into this class.
- *Not defined* (X). Describes whether a class is not explicitly or partially defined.

Table 12 reports the results of applying the previous criteria on the selected context models. At the end of this table, the percentages obtained for each criteria are specified. As shown, none of the classes are covered 100% in the context models. Regarding context information classes, the most addressed class is the *Location* class representing 65% of the classes explicitly defined with or without further hierarchies, followed by *Time*, *Activity* and *Environment* with 50%, 48%, and 31% respectively.

The explicit definition of the remaining context information classes in the context models decreases considerably with respect to those discussed above. None of them is explicitly defined by more than 28% of the proposals: *Profile* is defined in 26% of the proposals followed by *Role* with 16%, and *States&Status* with 9%. However, they are covered in context models by defining some of their sub-characteristics, i.e. partially defined. The results show that proposals partially cover *Profile, Role* and *States&Status* classes in 64%, 40% and 36% respectively. The percentage of classes that were not defined (neither explicitly nor partially) in the analysed context models is slightly high and significant, especially in context information classes such as *States&Status* with 55%, followed by *Time* and *Role* with 46% and 43% respectively.

Regarding classes of entities, most of the proposals do not provide neither an explicit definition of the *Agent* class that represents entities such as Person, Organism, etc., nor *Resource* that represent Services, Devices, etc. In fact, only 19% and 15% of the contributions explicitly define the *Agent* and *Resource* classes respectively with or without further hierarchies. Other 61% and 69% of the proposals, respectively, do partially define them, i.e., define some of their sub-characteristics. Finally, 37% of the proposals do not specify any entity and therefore, they prefer to characterize only context information that can be applied to different entities beyond focused on specific ones. Note that these classes (context information and entities) were previously selected by means of matches in the proposals (see Table 11), i.e. several classes are completely absent in a considerable set of context models, especially those that are specific in a domain.

Table 12. Context information classes and entities coverage

Proposals	**Agent**	**Resource**	**Activity**	**Time**	**Enviro nment**	**Location**	**Profile**	**Role**	**States & Status**
Henricksen et al. [66]	P	P*	X	X	X	X	X	X	X
Held et al. [67]	P	P*	X	X	✓	X	✓	X	X
Tarasewich [68]	P	P	✓*	✓*	✓*	✓	P*	X	✓*
Ghannem et al. [69]	P	P*	X	X	✓	P*	X	X	X
Mok and Min [70]	X	X	✓	✓	X	✓	P	X	X
Chen et al. [17]	✓	P*	P*	✓	P	✓	✓	X	X
Cao et al. [12]	P	P	✓	✓	P	✓	P*	P	X
Devaraju and Hoh [71]	✓*	✓*	✓	X	P	✓	P*	P	P
Pietschmann et al. [72]	✓*	P*	✓*	✓*	X	✓*	✓	✓	✓
Cadenas et al. [73]	P	P*	X	✓	✓	✓	P*	✓	P
Rubio et al. [74]	P	X	P	✓	✓*	✓*	P*	P	X
Cabrera et al. [26]	X	✓	✓	✓	✓	✓	✓	✓	X
Gu et al. [6]	✓	P*	✓*	X	P	✓*	P*	X	X
Hofer et al. [78]	P	P*	X	✓	X	✓	P*	X	X
Preuveneers et al. [79]	P	✓*	✓	✓	✓*	✓*	✓*	✓	P*
Heckmann [80]	P	P	X	X	P	P	P*	✓	P*
Chen et al. [81]	✓*	P*	✓*	X	X	✓*	P*	✓*	X
Xynogalas et al. [82]	P	P*	X	✓	P	✓*	✓	X	✓*
Bu et al. [83]	✓*	P*	X	X	X	X	P*	X	X
Kim and Choi [84]	✓*	P*	✓	✓	P	✓*	P*	X	X
Neto and Pimentel [85]	✓*	P*	✓*	✓*	P	✓*	✓*	✓	X
Park and Kwon [86]	P*	P*	✓	✓	✓	✓	P	✓	X
Ngo et al. [88]	✓*	P*	✓*	✓*	✓*	✓*	✓*	X	X
Bradley and Dunlop [89]	P*	P*	P*	P	✓	P	P	P	P
Kritsotakis et al. [90]	P*	P*	P	X	X	✓*	✓	P*	P*
Sheng et al. [91]	P*	P*	✓*	X	P	✓*	✓	X	X
Lee and Kwon [93]	X	P	X	X	P*	X	X	X	P
Chaker et al. [94]	P	X	P	X	✓	X	P*	P	P
Hervás et al. [96]	P*	P*	P*	X	P*	✓*	✓	✓	✓
Najar et al. [47]	✓*	✓*	P*	✓	P	✓	P*	✓	X
Prekop et al. [97]	✓	✓*	✓*	X	X	X	P	X	P
Khedr and Karmouch [98]	✓*	P*	✓	✓	P	✓*	✓	✓	P
Gong et al. [99]	P*	✓*	P	X	X	X	P*	P*	X
Villegas and Müller [100]	✓*	P*	✓*	✓*	P	✓*	P*	X	P
Strassner et al. [101]	✓*	✓	✓*	✓*	P	✓*	P*	P	X
Zimmermann et al. [102]	✓*	P*	✓	✓	P	✓	P*	P	P*
Shen and Cheng [103]	✓	✓*	✓*	✓	✓	✓*	P*	P*	X
Kayes et al. [104]	✓*	✓	P	✓	P	✓	✓	✓	✓
Śliwa and Gleba [105]	P	P*	P*	X	X	X	✓	P	X
Krömker and Wienken [106]	P	X	X	X	P*	✓	P	P	X
Aly et al. [107]	P	P*	X	X	X	✓*	X	X	X
Khemaja and Buendía [108]	P*	P	✓*	✓	P*	✓*	P	P*	X
Hoque et al. [109]	P*	P*	✓*	✓*	P*	✓*	X	P*	X
Hameurlaine et al. [110]	X	P*	X	✓	P*	✓	✓*	P*	X

Proposals	Agent	Resource	Activity	Time	Environment	Location	Profile	Role	States & Status
Khouja and Juiz [111]	P	✓	X	X	X	X	X	P*	X
Liang-Liang [112]	X	✓	X	X	X	X	P*	X	X
Saidani et al. [113]	P	✓*	P	✓	✓*	✓	P*	✓	X
Restrepo et al. [114]	X	P	P	P	P*	P	P	P	P
Lee et al. [115]	X	X	P*	✓	P*	P*	X	X	P
Xue et al. [116]	X	P*	X	X	P*	P*	P	X	✓
Wang and Tang [117]	✓	✓*	X	X	X	X	P*	X	P
Sohn et al. [118]	P	X	X	✓	✓*	P	✓	X	X
Yus et al. [119]	P	X	✓*	X	X	✓*	X	X	X
Choi et al. [120]	X	P*	X	P*	✓*	✓*	X	X	P
Sotsenko et al. [121]	X	P*	✓	✓	✓*	P*	P*	X	P
Li and Chen [122]	X	X	X	X	X	P	P*	✓	X
Chellouche et al. [123]	P	P*	P	P	✓	X	✓*	P	X
Zhang et al. [124]	X	P*	P*	✓*	X	✓*	P*	X	P
Berri [125]	P	P*	✓*	✓	X	✓*	P*	✓	X
Li et al. [126]	P	P*	✓	✓	✓*	✓	X	X	X
Katsumata [127]	P	X	✓	P	X	X	P	X	X
Da et al. [128]	X	P	X	X	X	P	P*	X	X
Benlamri and Zhang [129]	P	P*	✓*	X	✓*	✓	P*	P	X
Miraoui [130]	P	✓*	X	X	X	X	P*	P	X
Rahman et al. [131]	X	X	✓*	✓*	X	✓*	P*	✓*	X
Diallo et al. [132]	X	✓	✓	X	✓	✓	P*	P	P
Maalej et al. [133]	P	P*	✓	✓	✓	✓	✓	P*	X
Furno and Zimeo [134]	✓*	X	X	✓	P	✓*	P	✓	X
Novakovic and Huemer [135]	P	X	✓*	X	P	P	P*	X	X
Ai et al. [136]	P*	P	X	✓*	P*	✓*	P*	X	P
Boudaa et al. [137]	✓	P*	✓	✓	✓*	✓	✓	X	X
Kim et al. [138]	P*	P*	✓*	✓*	X	X	P*	X	P
Fissaa et al. [139]	P	P*	X	✓	P	✓	P*	P	P*
Zouhaier et al. [140]	P	P	✓	X	✓	X	P*	P	X
Kabir et al. [141]	P	X	P	X	X	X	P*	P	✓
Paul.and Wachsmuth [142]	P	X	✓	X	P	✓	P	X	X
Hu et al. [143]	P	X	✓	✓	P	✓	P*	P	X
Copeland and Crespi [144]	X	X	✓	✓	P	P	P*	X	X
Wong et al. [145]	✓	P*	✓*	X	X	✓*	P*	P	X
Jianfeng and Dong [146]	P*	P*	X	✓*	P*	✓	P	X	X
Madkour and Maach [147]	P	P*	✓	X	P*	✓	✓	P	P*
Jang and Choi [148]	P	✓*	✓*	✓*	P	✓*	✓*	X	X
Gilman et al. [149]	P	P*	P	✓	P*	✓	✓	P	P*
Wusheng et al. [150]	P*	P	✓	X	P	✓	P*	P	P
Li [151]	P*	P*	✓*	✓	P*	✓*	P*	P	X
Kim et al. [152]	P*	P	P	X	✓	P	P*	P	X
Changboka et al. [153]	X	✓	✓	✓*	X	X	✓*	P	X
Durán et al. [154]	P	X	X	X	✓*	✓	✓*	P	✓
Tong et al. [155]	P	P*	X	X	P	✓	P*	X	P
Luo et al. [156]	P	P	X	✓	P	X	P*	X	X
Yi et al. [157]	P	P*	P	X	✓	X	P*	P	P
Sun et al. [158]	P	P*	✓	X	✓*	X	P*	P	P
Sigg and David [159]	P*	P*	✓*	✓*	✓*	✓*	✓	✓	P*
Park et al. [160]	P	P*	✓	X	P	✓	P*	X	P
Zainol and Nakata [161]	P*	P*	✓	X	P*	✓	✓	X	P
Saeedi et al.[162]	P	✓	✓*	X	X	✓*	P*	X	X
Bandara et al. [163]	X	P*	X	X	X	X	P*	X	P
Liu et al. [164]	P	P*	P	✓	✓*	✓*	P	X	✓*
Hu and Li [165]	P	P*	X	✓	✓*	✓	P*	P*	P*
Cioara et al. [166]	✓	✓*	X	X	P	P	P*	P	X
Cao et al. [167]	P*	P*	✓	X	P	✓	P*	X	P
Lamsfus et al. [168]	X	P*	✓	✓	P	✓	P*	P	X

Proposals	Agent	Resource	Activity	Time	Environment	Location	Profile	Role	States & Status
Hur et al. [169]	P	P	✓	X	P	✓	✓	X	X
Coma et al. [170]	X	X	P	✓*	P	P	P*	P	X
Tan et al. [171]	P*	P*	✓*	✓	✓*	✓*	✓	✓	✓
Sun et al. [172]	P	P*	P	✓	✓*	✓*	P*	P	P*
Ouyang et al. [173]	✓*	P*	P	X	✓*	X	P*	P	X
Ko and Sim [174]	X	P*	✓*	X	X	✓*	✓	X	X
Chang et al. [175]	X	X	X	X	X	X	X	X	✓*
Ghadiri et al. [176]	P	P*	X	✓	P	✓	P*	X	P*
Erfianto et al. [177]	P*	P*	✓	X	P	✓	P*	X	X
Qin et al. [178]	P*	P*	✓*	✓	✓	✓	P	P	X
Lee and Meier [179]	X	X	X	✓	P	✓	P	X	P
Mei et al. [180]	X	✓	X	X	P	X	P	P	X
Paganelli et al. [181]	✓*	P*	✓	X	✓	✓	P*	P	X
Hwang et al. [182]	P*	P*	P*	✓*	P*	✓	✓	P*	P
Choi and Yoon [183]	P*	P*	✓	✓	✓	✓	✓	P	X
Hwang et al. [184]	P	X	P	X	P*	✓	✓	P	X
Jrad et al. [185]	X	P	X	X	X	✓	P*	X	P
Pessoa et al. [186]	P	P*	✓*	X	X	✓	X	X	X
Hamdeh and Ma [187]	P	P*	✓*	✓	X	✓*	X	X	P*
Strimpakou et al. [188]	✓	P*	✓*	✓	X	✓	P*	X	P*
Sheng et al. [189]	X	P	P	X	P*	P	P*	P	P
Wang et al. [190]	P	P*	P	X	P	P	P	P	P
Yang et al. [191]	P*	P*	P	✓	✓*	✓*	✓*	P	P*
Kranenburg et al. [192]	P	P*	P	X	✓	P	P	P	P
Chaari et al. [193]	P	✓	✓	X	P	✓	P*	X	X
Ou et al. [194]	✓*	P*	✓*	✓	P	✓*	P*	P	X
Almeida et al. [195]	P	X	P	P	P	✓	✓*	P	P*
Hu and Moore [196]	P	P	✓	✓	P	P	P	✓	✓*
Amundsen and Eliass. [197]	X	P*	X	X	X	X	P*	X	P*
Hong et al. [198]	P*	P*	✓	✓	✓*	✓*	P	X	X
Kalyan et al. [199]	✓	P*	P	✓	X	✓*	P*	P	P
Go and Sohn [200]	P*	P*	X	X	✓	P	P	P	P
Zacarias et al. [201]	P	X	P	✓	X	X	X	✓	X
Ko et al. [202]	P	P*	✓	X	✓*	X	✓	X	P*
Kaenampor. and Neill [203]	P*	P*	P	✓	P	✓	✓	✓	X
Korpipaa et al. [4]	P	P*	✓	✓	✓*	✓	P*	X	X
✓	**7%**	**7%**	**26%**	**38%**	**14%**	**36%**	**19%**	**15%**	**5%**
✓*	**12%**	**8%**	**22%**	**12%**	**17%**	**29%**	**7%**	**1%**	**4%**
P	**42%**	**12%**	**18%**	**4%**	**30%**	**12%**	**15%**	**33%**	**24%**
P*	**19%**	**57%**	**6%**	**1%**	**13%**	**3%**	**49%**	**7%**	**12%**
X	**20%**	**17%**	**28%**	**46%**	**27%**	**20%**	**10%**	**43%**	**55%**

We also identified the target domain of the different proposals, since it has an influence in the proposed vocabulary. The proposals targeted 16 different domains, being the most referenced ones: 1) services in Smart Cities and the Internet of Things (28 proposals, 20.0%) covering services in smart home, smart parking, smart campus, smart agents, public transportation, etc.; 2) software services (20 proposals, 14.5%) covering web service composition, discovery and adaptation, service provisioning and consumption and common services (e.g., email services); 3) business processes from the point of view of services (15 proposals, 10.9%) covering BPM, WoT business environment, business to business, etc.; 4) e-health services (15 proposals, 10.9%) covering healthcare services, health monitoring, etc.; 5) mobile computing (12 proposals, 8.7%) covering user interface tailored for disable users, social-aware applications, etc.; 6) generic models for services in the ubiquitous computing (9 proposals, 6.5%) covering middleware frameworks, social context. From the 138 proposals, 64 of them (46.4%) were generic proposals that were enlarged to demonstrate their extensibility in some domains specified.

2.3.4 RQ1.4. What are the most consolidated classes of context information and entities in ontology-based context models?

The goal in this research question is to analyze and evaluate the provided definitions of classes of context information and entities, with the objective of providing a set of definitions semantically coherent and generic enough for grounding context knowledge with lower levels of abstraction. For this goal, we analyzed the most consolidated definitions for classes of context information and entities in the selected context models. More precisely, we focused on the first level classes, which are the most frequent ones in the surveyed context models, and their synonyms (see Table 10). Hence, in order to obtain accurate and significant results, the analysis focused on clustering definitions that represent a generic description of a set of classes that share an exact or similar meaning (synonyms) without being ambiguous or highly abstract. Next, we sought and proposed, by considering the retrieved definitions, a definition taken from WordNet[5] [204] with the aim of decreasing misunderstandings of the clustered definitions, and therefore consolidating and unifying the generic view of such definitions. Our intention was to specify generic definitions that might fundament and cover several classes, rather than providing the "best" definition of classes. As an example, consider the following case of the *Agent* class:

- Def.1. "*Both computational entities and human users can be modelled as agents*" [17];
- Def.2. "*Represents the users*" [47];
- Def.3. "*Agent who performs the activity*" [97];
- Def.4. "*It provides information about the identities of the entities in the environment*" [98];
- WordNet. *"**A representative who acts on behalf of other persons or organizations**"* [204];

As it can be seen, the definitions follow certain similarity to each other. However, some of them are more generic than others. Using WordNet as stated above, we clarify the meaning of Agent in our ontology. From this perspective, we are able to say that an Agent can be a computational entity, human, user, etc., that performs an activity and that might acts on behalf of other agents.

Classes with larger hierarchical depth provide very specific definitions that we did not consider in this analysis. However, part of our criteria to group generic definitions was that from these definitions, specific ones could be instantiated. Given the example above, we considered that the compilation between the clustered definitions and WordNet provides a consolidated and unified vision of entities in the environment.

Table 13 summarizes the results of this research question. As it can be seen, in most of the first level classes we found at least one definition that might be classified generic enough as to include other classes and their corresponding definitions. In this sense, the classes with more number of generic definitions identified are *Time* and *Location* with 4 definitions each covering around 22 and 25 classes of other proposals, respectively, followed by *Activity* with 2 definitions covering 20 context classes of other proposals. Contrary to them, *Profile, Role* and *States and status* were the classes with less number of generic definitions; this means that most of the classes defined regarding to them, refer to a particular domain or describe a particular entity. The results obtained show a very low number of generic definitions considering that we have analyzed 2.756 context classes. This means that most of the context models have been applied on specific scenarios of the context-aware area from the perspective of service-oriented computing, defining context classes domain-dependent that were not linked with upper-level ontologies to provide a rich semantic.

[5] WordNet has been used for a long time as a source of nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Table 13. Most consolidated definitions of classes of context information and entities

Class	Definition	Definition coverage
Agent	1) *"Entities that perform some action (e.g. people, group and organizations)"* [85]. 2) *"Provides information about the identities of the entities in the environment"* [98]. 3) *"**A representative who acts on behalf of other persons or organizations**"* [204].	Chen et al. [17]; Cadenas et al. [73]; Hofer et al. [78]; Chen et al.[81]; Ngo et al. [88] Najar et al. [47]; Prekop and Burnett [97]; Strassner et al. [101]; Aly et al. [107]; Cioara et al. [166]; Ghadiri et al. [176]; Paganelli et al. [181]
Resource	1) *"Resources express people, services or tools that can be used in to achieve a particular goal or task"* [99]. 2) *"Describes anything used to perform the activity. Resources needed by the activity"* [97]. 3) "**A source of aid or support that may be drawn upon when needed"** [204].	Strassner et al. [101]; Shen and Cheng [103]; Khouja and Juiz [111]; Liang-Liang [112]; Saidani et al. [113]; Miraoui [130]; Cioara et al. [166]; Mei et al. [180]
Activity	1) *"The activity context of the agents"* [81]. 2) *"Actions and tasks performed by an object"* [100]. 3) *"**Any specific behaviour**"* [204].	Tarasewich [68]; Gu et al. [6]; Hofer et al. [78]; Preuveneers et al. [79]; Park and Kwon [86]; Ngo et al. [88]; Strassner et al. [101]; Zimmermann et al. [102]; Khemaja and Buendía [108]; Hoque et al. [109]; Benlamri and Zhang [129]; Hu et al. [143]; Madkour and Maach [147]; Gilman et al. [149]; Wusheng et al. [150]; Park et al. [160]; Erfianto et al. [177]; Paganelli et al. [181]; Choi and Yoon [183]; Pessoa et al. [186]; Strimpakou et al. [188]
Time	1) *"Involves timeline of past, present, and future. This allows for a record of past context, which can be used for comparison to the current context or for predicting future context"* [68]. 2) *"Describes temporal properties and temporal relations between different events"* [12]. 3) *"Models information that is purely temporal in nature"* [101]. 4) *"Temporal concepts and properties common to any formalization of time"* [17]. 5) *"T**he continuum of experience in which events pass from the future through the present to the past**"* [204].	Cadenas et al. [73]; Rubio et al. [74]; Xynogalas et al. [82]; Park and Kwon [86]; Ngo et al. [88]; Najar et al. [47]; Villegas and Müller [100]; Zimmermann et al. [102]; Khemaja and Buendía [108]; Hoque et al. [109]; Ai et al. [136]; Boudaa et al. [137]; Liu et al. [164]; Ghadiri et al. [176]; Lee and Meier [179]; Hwang et al. [182]; Choi and Yoon [183]; Strimpakou et al. [188]; Kalyan et al. [199]; Zacarias et al. [201]; Kaenampornpan and Neill [203]
Environment	1) *"Consists in modeling all the environmental factors"* [94]. 2) *"**The totality of surrounding conditions**"* [204].	Tarasewich [68]; Preuveneers et al. [79]; Park and Kwon [86]; Krömker and Wienken [106]; Khemaja and Buendía [108]; Hoque et al. [109]; Boudaa et al. [137]; Sun et al. [158]; Zainol and Nakata [161]; Liu et al. [164]; Choi and Yoon [183]
Location	1) *"By location context, we mean a collection of dynamic knowledge that describes the location of an agent"* [81]. 2) *"Provides information about the location of entities as identified and detected by location-type sensors"* [98]. 3) *"A location may be described as an absolute location, meaning the exact location of something, or as a relative*	Tarasewich [68]; Chen et al. [17]; Cao et al [12]; Cadenas et al. [73]; Gu et al. [6]; Hofer et al. [78]; Xynogalas et al. [82]; Ngo et al. [88]; Najar et al. [47]; Krömker, and Wienken [106]; Boudaa et al. [137]; Madkour and Maach [147]; Wusheng et al. [150]; Durán et al. [154]; Ghadiri et al. [176]; Erfianto et al. [177]; Lee and Meier

	location, meaning the location of something relative to something else" [102]. 4) *"Represents the abstraction of a physical location, which has two subclasses including LogicalLocation and PhysicalLocation"* [103]. 5) *"**A point or extent in space**"* [204].	[179]; Paganelli et al. [181]; Choi and Yoon [183]; Jrad et al. [185]; Pessoa et al. [186]; Strimpakou et al. [188]
Profile	1) *"Profile Info of entities"* [104]. 2) *"**An outline of something**"* [204].	Held et al. [67]; Xynogalas et al. [82]; Ngo et al. [88]; Hervás et al. [96]; Śliwa and Gleba [105]; Maalej et al. [133]; Gilman et al. [149]; Zainol and Nakata [161]; Hwang et al. [182]; Choi and Yoon [183]; Hwang et al. [184]; Almeida et al. [195].
Role	1) *"Role of an agent can be used to characterize the intention of the agent"* [81]. 2) *"**The actions and activities assigned to or required or expected of a person or group**"* [204].	Hofer et al. [78]; Preuveneers et al. [79]; Park and Kwon [86]; Khouja and Juiz [111]; Saidani et al. [113]; Rahman et al. [131]; Kabir et al. [141]; Hu and Moore [196]; Zacarias et al. [201]; Najar et al. [47]; Kaenampornpan and Neill [203]
States and Status	1) *"Status Info of entities"* [104]. 2) *"**A state at a particular time**"* [204].	Tarasewich [68]; Xynogalas et al. [82]; Hervás et al. [96]; Kabir et al. [141]; Hu et al. [143]; Durán et al. [154]; Liu et al. [164]

2.4 Discussion of the systematic mapping

In this section, we discuss the observations gathered when answering each research question specified in this review.

RQ1.1. *What is the chronological overview of the research done so far in ontology-based context models?* The distribution of papers over the years (see Fig. 3) shows that ontology-based context modeling is a lively area. In fact, if we consider the selected papers (i.e., papers that fulfilled the inclusion criteria), 2014 experienced a dramatic growth becoming even the year with more selected works in the period of this review. In our opinion, the reasons for context being of such interest for the research community are twofold. On the one hand, the great importance of context in modern service-oriented computing and software systems (Smart Cities, IoT, self-adaptive systems, etc.) that demand deep knowledge on context in order to deliver good solutions to the citizens. On the other hand, the absence of a standard for context modeling (the most widespread proposal, SOUPA cannot be considered as such) triggers work trying to contribute in this direction.

The analysis per provenance shows an unsurprising dominance of university papers. This observation is commonplace when surveying the state of the art based exclusively in academic papers. Still, it is remarkable that such dominance is very high. If we compare to related fields, we found similar numbers in a former SM on ontologies for quality of service [205] where the percentage of industry papers was 9%, only slightly higher than the 5% reported in this contribution of the thesis. It is worth remarking that the cited SM on quality of service found also a 4% of proposals coming from organizations like W3C or OASIS, which is missing in the context field. If we compare with other SMs in software engineering, we find percentages of industry papers even greater than 30% [206][207][208].

We also distributed the context models chronologically exploring their relationships with the aim of identifying the most influencing, influenced and consolidated contributions. The most remarkable observation from this chronological analysis is that current context models have had a limited impact on each other until now: only 8% of the proposals have influenced other proposals (only two of them, CONON and SOUPA, influenced more than 5 other approaches). Remarkably, 59% of the proposals were formulated from scratch. Also, the transitivity analysis showed that their impact didn't propagate much to subsequent models. We consider this as an indicator that the community is still not mature enough in spite of the large amount of existing proposals. On the contrary, as a positive observation, we noticed the influence of classical ontological works in some context models proposals (e.g., FOAF, OpenCyc, DAML).

Considering such findings together, they suggest the need of development of an ontology-based context model that can act as a basis of prospective proposals and that can provide a unified and consolidated body of knowledge on context modeling in the field of service-oriented computing. Such context model should comprise context knowledge pieces from the perspective of service-oriented computing and appropriate parts of the ontologies (context-oriented or not) that were the basis of different context models found in this SM. We believe that the chronological analysis shown in this SM and summarized in Fig. 6 can be of great help to this future work. For instance, it shows which proposals have been defined considering existing work, or which proposals exert more influences. For its wide adoption, industry and/or standard bodies should be involved more than currently are.

RQ1.2. *What are the characteristics of the proposed ontology-based context models?* As the results of this research question point out, most of the models include knowledge pieces that are introduced but not defined, neither in the paper itself nor through links to external documentation. In more detail, the correlations found among these knowledge pieces reflect deficiency of completeness, clarity, consistency, expandability, robustness and coherence. This situation was mostly observed in models with big size, i.e. models providing several classes and higher depth level, and in models that integrate or merge different ontologies. Instead, models with smaller size reflect more coherence among the knowledge pieces specified therein. It could be thought that this fact comes from the inherent limitation of presenting research results in the limited space of a scientific paper but as commented above, the information was not available through external links either. The findings of this research questions also show that most of the relationships among context information classes represent a basic subclass relation and only a small set of proposals provide domain-dependent relationships. Both facts together claim for a more systematic and thorough presentation of ontology-based context models as a pre-requisite for their wide adoption.

The analysis of correlations among the number N of context information classes (which is used as indicator of the size of the models) and the rest of entity types shows an interesting fact: while there is an approximate linear relationship among N and the depth levels of the hierarchy (see Fig. 7), such a relationship does not exist with the others: 79% of the models have less than 10 datatype properties and 83% of them have less than 10 object properties, in both cases regardless of the value of N (which ranges from 3 to more than 40). In other words, contrary to what we expected at least in object properties, the size of the analyzed datatype and object properties is not correlated with the size of classes specified. We expected 100% of completeness of object properties at least at abstract levels of the model, i.e., the first or second level of the hierarchy from which it is possible to reason and query facts and assertions that involve lower level concepts (specific classes of context information and entities) that do not have any relation with other class. Going into the details of the papers, the main reason is that these papers usually focus on only one part of the domain model in order to develop an example, while the rest of the model remains unexplored and poorly documented, as stated in the previous paragraph.

In more detail to datatype properties, although the literature does not state the most suitable number of datatype properties to link a class to a data value, we consider that the number of datatype properties provided in the reviewed context models is too low. This fact may indicate that most of the context models were designed to support complex context-aware applications or systems that do not manage simple specifications or value restrictions, i.e., describing certain context by using properties for which the value is a XML schema datatype (e.g., describing the profile of a person as an expected triple "Person hasName xsd:String") or trigger an event given certain value restriction (e.g., if a person hasAge xsd:Int greaterThan "18" then the person is classified as an adult). At this respect, most of the context models employed in context-aware services supporting such type of descriptions or restrictions were those focused on profiles and preferences of the user; have reused ontologies such as FOAF; describe the location of someone/something by direct values of latitude and longitude; etc. The retrieved evidence clearly leads to conclude that most of the context information classes were not linked to data values by means of datatype properties; therefore, although this situation can be caused for modelling decisions, we observe a lack to identify datatype properties consistent to the domain.

In more detail to object properties, although a set of object properties may be related only to a class in a model as also happens with the datatype properties, this correlation was expected, i.e. the higher number of classes is, the higher the number of object properties should have been. According to common literature on ontology building [58][95][209], one of the main modeling components are the object properties that represent a type of association between concepts of the domain. However, the object properties proposed in the models are too low since most of them were developed with very low range of object properties, i.e., 82.6% of the models have presented less than 10 object properties.

We also observed that although the difference of the total number between both properties (743 object and 694 datatypes) is not representative, most of the assessed contributions were focused on object properties. We could identify that it is because the following reasons: some context models have been designed to characterize their properties as transitive or symmetric, and only the object properties provide this type of characteristics; reasoning engines can be able to exploit the expressivity of object properties increasing the reasoning capabilities of the ontologies; powerful inferences that can be obtained from the object properties (e.g. if certain location X is affected by the pollution of the environment, and a person Y is located at X, then the pollution of the environment also affects the person Y); etc. Most of the object properties used in context-aware services are those related to the Location class (e.g., isLocatedAt, isLocatedNear, etc.), Service class (e.g., usesService, providesService), User class (e.g., hasRole, hasProfile, etc.), Time class (e.g., before, after, to, from, etc.), and so on.

Regarding the completeness of ontology-based context models, we observed that the definitions of model components are very incomplete in general. At the level of classes, only a small set of proposals (21%) have a unique and consistent definition for more than 90% of their context information classes, and 63% of them have a completeness definition below 60%. The situation does not improve if we focus on datatype and object properties, e.g. only 4% of the proposals include datatype properties at least in 80% of their classes, and only 9% of them define object properties among their classes.

All in all, we can conclude that the level of detail on the context models given by the current proposals is not optimal, hampering their adoption. In fact, going back to RQ1.1, this lack of detail can be one of the reasons for the low impact of existing context models. If we go a bit further, we observe that ontologies with a more generic scope and rich expressiveness (i.e. classes of context information duly interrelated by means of data type and object properties that consider a robust specification of their characteristics –functional, symmetric, reflexive, etc.– and in some cases also involving rules), were ontologies that although have presence and importance for the contributions in the review, were not widely reused. Instead, context modelers seem to prefer the definition of lightweight ontologies, i.e. ontologies with low level of detail considering a small set of context information classes, basic structure of the model (commonly a context hierarchy), poor characterization of data type and object properties, etc. For this reason, ontologies such as SOUPA and CONON are two of the most considered in different context modelling contributions. From our point of view, this consideration is not bad, but would be interesting that the modelers can combine both perspectives in a way that the ontology can be reused minimizing the complexity of this process of reusing. In any case, maintaining ontologies in a basic structure with basic relationships seems to be enough to operate with them.

Following the development of a context model that can act as a basis of prospective proposals and considering the findings of this RQ, a future context model can comprise or avoid the following source of knowledge: 1) Given that the most of the proposals have been created from scratch, the modeler can avoid it and justify and encourage the reuse of models as suggested by the existing building methodologies of ontologies; 2) Given that a depth level indicates the level of abstraction of a class, the modeler can identify the models that provide the most abstract or domain-dependent classes of context in order to unify the abstraction of the vocabulary; 3) Given that most of the proposals do not provide the definitions of the context knowledge pieces, the modeler should consider to increase the reusability of the model by providing this kind of details; etc.

RQ1.3. *Which classes of context information and entities are the most addressed in the ontology-based context models?* According to our study and after processing synonymous terms, the most addressed context information classes explicitly defined are location (appearing in 65% of the proposals), time (50%), activity (48%) and environment (31%). Their wide coverage is not surprising since obviously from the Location of an entity different kind of context information classes can be derived or deduced, as happens with Time, Activity, and Environment which allows to reason about different dimensions of time, activities performed by or around entities, as well as reasoning about environment issues that could affect positively or negatively certain entities in an implicit or explicit interaction process given a Location.

Regarding entities, we found agent (19%) and resource (15%) as the most frequent ones. These numbers show that surveyed papers focus more on context information itself than on the entities to which this information is bound. This fact is related to the modelling styles employed in a model. We observed that most of the designers have preferred not clearly separate the entities that are being characterized by means of context information. In this regard, some models provide a more generic scope by modelling only context information without characterizing any specific entity. Such modelling styles allowed us to identify three patterns of combination between entities and context information classes (what we called entity-dependent, entity-independent and context-sensitive, see Fig. 12). As it was observed, the most used is the context-sensitive, which from our perception, when this modelling style employs a term to define both an entity and different types of context information classes, has the problem of data redundancy and coherence in a higher percentage than the others (e.g. classes also defined as instances or properties).

The results also indicate that other classes such as profile and role appear with less frequency than the described above. It means that a small set of contributions provide axioms to describe the application domains of these context information classes. Furthermore, most of the axioms generated through these classes were focused on the following applications domains: most of the environment classes appearing in 33% of the proposals were focused on climatological parameters, most of the 27% of the profile classes were focused on human profiles, and most of the 17% of the role classes were focused on roles of a person. It indicates that the scientific community can explore for future context models, other domains of these classes such as the social environment, different profiles and roles of resources, among others.

As it was observed, all the context information classes and entities oscillate in different degrees of definitions and further elaboration (i.e., decomposition into simpler concepts) indicating the need to unify and consolidate the vocabulary and context modelling as also happen in many other areas. At this respect, we have observed that most of the vocabulary previously assessed has been used from a domain perspective, and the resulting domain ontologies developed have not been mapped with a middle-level, upper-level or foundational ontology. Similarly, the proposals that have reused or proposed an upper or middle level ontology to map domain ontologies presented 1) no mappings to foundational ontologies or any other context model proposal; and 2) mappings and reuses of different sources that lack of unification or standardization. This is one of the reasons for what we found several inconsistencies, i.e. several variants in synonyms, hierarchies and coverage of the context knowledge pieces. In fact, in a same domain we can find several lacks of a semantic unification.

Considering the findings of this RQ, a future context model can comprise or avoid the following source of knowledge: 1) Given that three types of patterns were identified, the modeler should considering the outlined benefits and drawbacks, from our point of view the entity-independent pattern allows modeling the entities that will play a role in a scenario and therefore, characterizing their situation can be more easy and the queries to the ontology to retrieve such situation by entity should be more expressive; 2) Given that there is a big inconsistency among the vocabulary used in existing contributions of context modelling, we have pointed out synonyms and hierarchies that should be considered to avoid semantic inconsistencies in the definition of the model; 3) The matches of the classes in different levels of a hierarchy indicate that existing models should be unified to provide a consistent hierarchy and better understanding of the model, it can be useful for domain ontologies with a bottom-up approach to be mapped with abstract classes; etc.

RQ1.4. *What are the most consolidated classes of context information and entities?* The lack of a standard and the diversity of context modelling contributions have prompted to the diversification and redefinition of different knowledge pieces of a model. Therefore, we analyzed the generality level of the definitions provided for describing the first level classes of the surveyed context models, particularly the most frequent ones. The results indicate that more than 60% of the contributions provide definitions entity-dependent or domain-dependent to describe a class (context or entity) that in most of the models belongs to the first level classes. It means that the scope of these classes that can be considered generics is reduced and therefore, the expandability of a model is affected. From this perspective, the Profile class is one of the classes in which most of its definitions depend on the domain or entity related. Instead, classes such as Location, Time and Activity were defined in a more generic form, i.e. most of their definitions were not related to a domain or specific entity. Thus, a class with these features should be more easily expanded.

Finally, a future context model should consider the most consolidated definitions of classes of context information and entities to unify the heterogeneous definitions of classes, from abstract levels toward domain levels all of them maintaining a consistent definition and therefore, increasing the reusability capabilities of either, the entire model or the context knowledge pieces specified.

To sum up, this mapping study has pointed out that there is a big lack in reusability, standardization, unification and consolidation of context knowledge pieces for context modeling. We identified different inconsistencies regarding context information classes and entities, one of the most important issues to reuse vocabulary of the context models analyzed is the lack of homogeneity among the classes, the resources provided and shortage of definitions. Furthermore, as it happens in many other areas, it does not exist a single context model agreed by the scientific community; instead several proposals have been presented for specific purposes. These results show the need to consolidate the context knowledge already provided. As a first step in this direction, we have made the exercise of defining the classes of context information and entities obtained from the research questions 1.3 and 1.4. Fig. 13 shows the resulting context taxonomy, corresponding to a high level hierarchy view of the most addressed classes in the surveyed context models. Note that we are not providing a complete class taxonomy, but a basic taxonomy of high-level context classes feasible for extending and reusing the model. We consider that the structure provided (organized according to the entity-independent pattern, see Fig. 12-B) is more generic and allows characterizing any entity with different context information classes. Hence, in this case, the surroundings of resources and agents can be modelled using the context information classes of the taxonomy. An extended report of classes (context information and entities) can be found at [59].

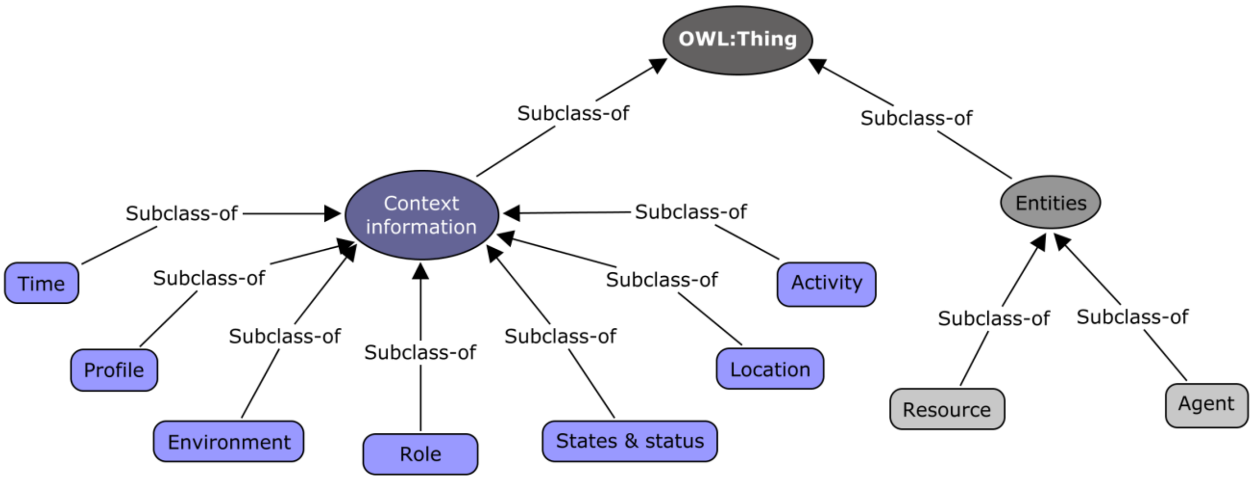


Fig. 13 Basic taxonomy of high level classes that should contain a context model

2.5 Threats to validity

This section addresses both the aspects of the research process that might represent threats to validity and the actions performed to mitigate those risks. To do so, we have identified and evaluated validity threats following the common classification of validity concerns: *construct validity*, *internal validity*, *external validity* and *conclusion validity*, discussed in [210] and covered in the following subsections.

2.5.1 Construct to validity

Risks: *1)* One of the inherent threats in any systematic review is that it does not guarantee the inclusion of all relevant works in the field. This represents a risk that might be caused by different reasons, such as: contributions in the field are not indexed on the selected database, the keywords defined in title and abstract of a contribution do not match with the keywords established in the search string, and so on. *2)* Threats beyond an accurate protocol comprising issues related to the paper, for instance, inaccurate abstracts.

Mitigation actions: *1)* We combined automatic and manual searches, considering different databases (Scopus, IEEE Xplore and ACM DL), specifying several relevant journals and conferences in the field, and studying accurately the keywords to use in the search. The identification of some basic sources (see Table 6) was useful since some of the conferences and journals that we considered as usual venues for the topic addressed in this review, had at least one edition not indexed in the selected databases, so that, we performed manual searches. It is worth remarking that, according to our experience, the number of papers in 2015 (and even 2014) may slightly increase in the future since digital libraries are not always up to date. However, we remind that the most critical venues have been manually checked until 2015, so that the missing papers should not be many and their relevance, limited. *2)* We included a final step of snowballing, as described in Section 2.2.5, assuring the inclusion of context models that have had a biggest impact in the field.

2.5.2 Internal validity

Risks: *1)* Several context models do not provide an accurate definition of the presented context information classes and entities. As shown in Section 2.3.2, some definitions are ambiguous, inconsistent or simply absent. This situation represents a big challenge when analyzing the coverage of context classes, and a subsequent threat to validity. *2)* Given the lack of a reference context model (formal standard) and considering that each context model provides its own hierarchical structure (see Fig. 12), we decided grouping synonyms analyzing each term of the proposals, verify matches and positions of each term grouped, and consolidate the information retrieved. This process can be considered by itself a threat to validity.

Mitigation actions: *1)* We analyzed the list of ill-defined context classes, establishing the following strategy: for those classes lacking of a definition, if there was a clear consensus of the definition in the state of the art, or the hierarchical path was self-explanatory (e.g. if the root of profile is a user class, profile obviously refers to context information class of a user), we recognized the meaning of the context class despite the lack of a definition; classes whose definition was ambiguous or inconsistent, but it didn't affect the categorization of context classes, they were considered in the analysis, otherwise, they were discarded. Following this criterion, around 60 context classes were discarded from more than 2000 classes analyzed, representing only the 3% of this quantity. *2)* To mitigate this risk, the process and specified criteria to address it were discussed and analyzed closely by experts in the area and the author of this thesis, consolidating in a systematic way the retrieved context knowledge.

2.5.3 External validity

Risks: *1)* Several context models were developed for specific purposes, delimiting their reusability in different cases and therefore forcing the creation of new context models starting from scratch.

Actions: *1)* The service perspective allowed to consider different perspectives such as user, provider, consumer, etc. from which several context models were built. Therefore, the resources consolidated encompass all the perspectives already provided in context modeling, especially, resources regarding context information vocabulary, object and datatype properties and definitions completeness. These resources are intended to be a knowledge source easy to reuse in different use cases.

2.5.4 Conclusion validity

Risks: *1)* Several surveys did not provide the guidelines addressed in the research, which limits to reproduce the research performed and therefore, the results can be different.

Actions: *1)* In this sense, we have explicitly described all the steps performed in the systematic mapping by detailing the procedure as defined in the systematic mapping protocol. Furthermore, the list of papers found and selected on each step is included at [59].

"If you wish to make an apple pie from scratch, you must first invent the universe"

- Carl Sagan

CHAPTER 3

How to model: building a context model

The effort in context representation seems to go along the way of providing a formal and well-defined context infrastructure. Context modelling plays such role of context representation according to a meaningful sense that is a relevant process into the context life cycle. For this purpose, different proposals on context modelling have emerged in order to define models for representing and structuring the context knowledge. However, as the review on context modelling summarized in Chapter 2 indicates, there are still several gaps to face in context modelling such as generic context models with capabilities of reusability and extensibility to deal with heterogeneous context information, unification, standardization and consolidation of the existing context knowledge, a well-defined data schema for improving its dissemination and storage, etc.

The aim of this contribution of the thesis is to build a context model that can cope the gaps pointed out above. For this purpose, different tasks were performed: (1) selecting, from the different patterns for modelling context, the most appropriate formalism to represent context knowledge, (2) developing the context model according to the basis of such formalism and the appropriate methods and techniques for building the context model, (3) specifying the relevant aspects to be considered from a service-centric perspective, and (4) considering all the findings related to the survey on context modelling conducted in Chapter 2.

The final result of this contribution is a context model in a form of a *Three-Level Context Ontology* for representing, reasoning and disseminating context. Specifically, the proposed *Three-Level Context Ontology* is intended to be easily reused, extended and adapted for specific or generic purposes covering several gaps on context modelling. Furthermore, the context model consolidates the context knowledge already available from a modular perspective yielding a clear schema of knowledge reutilization. For this purpose, we gathered and reconcile the most appropriate aspects and resources of existing contributions on context modelling to be integrated into unified, consolidated and well-defined modules. Summing up, we aim at building a proposal aligned with existing context ontologies, reusing ontological resources semantically well-defined acting as a pattern repeated in different proposals.

3.1 Methods and techniques for building the ontology

To systematically build the context model under the ontology-based approach, in this section the baseline methods and techniques to be considered in the development process of the context model are introduced.

3.1.1 Classification of the ontology

According to Gruber an ontology is an *"explicit specification of a conceptualization"* [211]. This definition well-known and highly referenced was complemented by Borst as *"a formal specification of a shared conceptualization"* [212]. Later, Corcho et al. [213] stated that a formal specification means that the ontology should be machine-readable. From this point of view, different designs and structures of ontologies have been proposed so far. Two usual criteria are: generality and expressiveness. Generality supports the adoption of a layered view of ontologies and has the purpose of specifying general classes towards top levels (abstraction) and more specific classes towards lower levels (granularity) [214]. Expressiveness indicates the level of detail of an ontology. Usually ontologies are classified into lightweight and heavyweight [213]. Whilst lightweight ontologies include concepts, taxonomies, relationships between concepts and properties that describe concepts, heavyweight ontologies add axioms and constraints. At this respect, the context model under the *ontology-based modelling* proposed in this thesis follows an ontology-layered view and a lightweight expressivity with the aim of providing a model easy to use and adapt in different use cases.

3.1.2 Ontology development process

Different comprehensive methodologies for developing ontologies have been proposed so far. These methodologies describe the stages and activities that should be performed to develop and maintain ontologies. Some of the most known and accepted methodologies found in the literature are those proposed by Fernández-López et al. (Methontology) [58][92], Uschold and King's [209], Grüninger and Fox's [215] and Staab et al. [216]. In the present thesis, we aligned the ontology building process considering the recommendations of these methodologies, especially from Methontology due to its evolving prototype life cycle that allows going back from any state to any other if some definition is missing or wrong.

3.1.3 Knowledge integration process

Commonly, the methodologies for developing ontologies specify an activity focused on the reuse of existing knowledge in order to avoid terminological ambiguities and improve the efficiency of the ontology building process. Since we have special interest in consolidating the existing context knowledge already defined in the literature, we focus on the reuse of existing ontologies. As stated by Pinto et al. [217], there are two different reuse processes: merge and integration. In a merge process, the ontology is built in one domain by reusing two or more different ontologies belonging to that domain. Hence, the source ontologies are unified into a single one, being usually difficult to keep traceability in the unified ontology, e.g. identifying which regions were taken from the merged ontologies and were left unchanged. In an integration process, the ontology is built in one domain reusing one or more ontologies in different domains (which may be related). Hence, the source ontologies are aggregated, combined and assembled together, to form the resulting ontology, possibly after reused ontologies have suffered some changes, such as extension, specialization or adaptation. In our approach, we have followed an integration approach because we are more interested on unifying modules than complete ontologies. For this purpose, we have adopted the integration process defined by Pinto and Martins [218] since they have compiled integration activities from different methodologies.

3.2 Specifications of the proposed context ontology

The context ontology proposed in this PhD thesis has the purpose of compiling, structuring and consolidating a generic body of context knowledge easy to reuse for characterizing the situation of different entities across multiple domains. Particularly, entities responsible for maintaining the value chain generated between them and services. At this respect, the knowledge acquisition carried out in Chapter 2, the brainstorming with service experts and the common literature of services (e.g. [21][219]) indicates that there are basic entities responsible for maintaining the value delivered through services such as provider, infrastructure, interaction means, consumer and third-party services.

From the perspective of the proposed context ontology, such entities have an explicit or implicit context that is critical to the health of the value delivered. Hence, the context of each entity can be specified as an abstract context composed by an aggregation of context information or context elements affecting one or more entities (see Fig. 14). From this perspective, the context model should be able to response questions related to entities participating in a process of service provisioning and consumption such as: Which are these entities? Which are the features of the entities that delimit the value delivered? What is the context information that can be used to characterize the situation of each entity? What is the context information that affects negatively the consumption of certain service? What is the context information that characterizes interaction means affecting positively or negatively the service consumption? And so on.

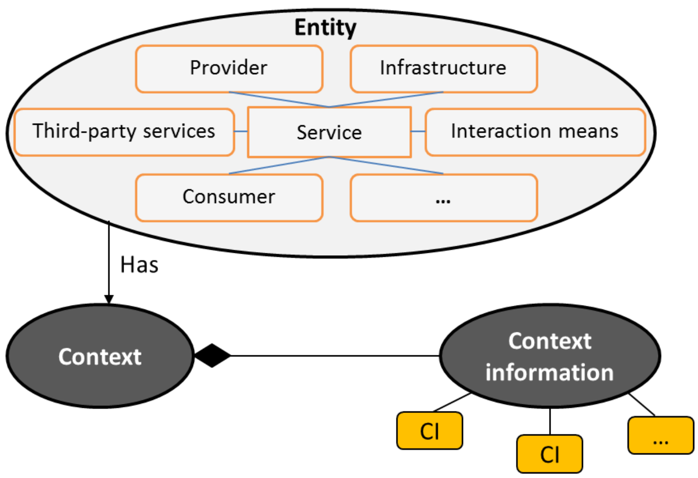


Fig. 14 Specifications from a service-centric perspective

The specifications of the proposed context ontology are thus stated as follows:

1) The domain of the ontology is the context than can be aligned with the definition given by Dey [3] who sates that *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*. Taking relevance, the context causing either positive or negative effects on entities involved in the value structure given by services.

2) The purpose of the ontology is to represent and provide a unified and consolidated context knowledge easy to be reused, extended or adapted. Required in context-aware services and for facing several gaps in context modelling such as dealing with heterogeneous context information, dependencies among the context knowledge represented in the models, a well-defined data schema for improving its dissemination and storage, and so on.

3) The scope of the context model is delimited by the representation of generic context knowledge that characterizes the situation of different entities involved in the value structure given by services, and those identified in the review on context modelling conducted in Chapter 2. The central idea is to visualize the context information that affects positively or negatively to an entity participant in the interaction given in a process. Hence, the model should provide the primitives need to model different context-aware service domains (e.g., the interaction between a person and smart parking service, maybe it is affected by social contexts such as strikes).

To achieve these specifications, the context model proposed in this PhD thesis is designed from the perspective of a three-level context ontology following an entity-independent pattern (see Fig. 12) in order to provide a generic model with different layers of abstraction that can be easy to reuse and extend.

3.3 3LConOnt: A three-level ontology for context modelling

3LConOnt is articulated around three-levels of detail for context modelling, namely upper, middle and lower level ontologies. This level-based modelling strategy has been adopted for covering the specifications previously detailed and some gaps found in the study on context modelling, particularly related to reusability issues and the lack of consolidated ontological resources. From this perspective, we consider the following main benefits:

- The model specifies vocabulary at three levels of detail, which facilitates its reusability [212]. Hence, an ontology designer is able to define and structure further vocabulary in the proper level of abstraction. Such capability is validated by extending the vocabulary proposed at each level of the model with the vocabulary taken from different smart service scenarios.
- The upper and middle levels of the model and all the modules of the middle-level ontology can be reused independently of the entire model. Hence, an ontology designer can integrate in an existing ontology the level of abstraction that is required without the dependency of other levels of the model. From the same perspective, an ontology designer can reuse any module of the middle-level ontology without inconsistencies since the semantic of each module is independent of each other (e.g., the module of time can be reused without the integration of other modules since it was formulated to represent only the semantic of time). Such capability is validated by illustrating into the Protégé tool[6] the reuse of a level of the model in an existing ontology.
- Lower-level ontologies and modules of the middle-level ontology can be integrated with other modules or domain-dependent ontologies maintaining the reasoning capabilities defined in upper levels. For this purpose, the ontology designer should follow a formal integration methodology as the presented in this part of the thesis to avoid inconsistencies in the generated axioms. Such capability is validated through the integration process of modules and levels of the ontology and by means of queries that involve the reasoning of the domain-dependent modules and upper levels.

For providing the required benefits and functionality of the model, the remainder of this chapter of the thesis focuses on the upper-level, middle-level and lower-level ontologies.

3.3.1 Upper-level context ontology

The *upper-level* provides high-level classes consolidated from the context models reviewed in the systematic mapping conducted in Chapter 2. The aim of this *upper-level* depicted in Fig. 13 is to provide a basic taxonomy of context classes that represents general context concepts like time, location, agent, etc., which are independent of any particular problem or domain. To populate and model the *upper-level ontology*, we have considered different aspects from the surveyed models. In particular: the most

[6] http://protege.stanford.edu/

addressed classes belonging to the first three levels of their proposed hierarchy; the definition and semantic description of these classes; common patterns identified through the proposed schemas; the alignment with foundational ontologies that were partially reused by the contributions; etc.

We also considered the study on modelling styles employed in a context model and the alignment with the definition of context given by Dey [3] (see Chapter 1). From the perspective of Dey, Person, Place, Object, User and an Application should be the general entities that participate in an interaction process, and context information the information used to characterize the situation of such entities. In our proposal, *the upper-level ontology* follows this logic and abstracts the mentioned entities and context information into different generic terms found by means the mapping study. For instance, we conceptualize and define an entity based on two big terms, Resource and Agent, where Resource is intended to provide the semantic needed to describe any resource such as an Object and an Application, and an Agent is intended to provide the semantic needed to describe any agent such as Person and User. The only variation in this alignment falls into the Place concept, according to the patterns found in the literature such concept beyond being an Entity it was represented as context information. Hence, such pattern is also considered in our proposal. With this consideration, we are able to say that the status of different entities is also affected by their location and another context information.

Summing up the previous considerations, the *upper-level ontology* (see Fig. 13) has been designed following a pattern that we called entity-independent in which entities and context information are clearly separated providing a more generic view of the model and its primitives (concepts and properties). Hence, entities into the Agent concept (e.g., Person, Organization, etc.) and Resource concept (e.g., Service, App, etc.) can be related, described or characterized through context information classes such as Profile, Activity, Environment, etc., that also can be extended by more specific modules or concepts. Hence, with the purpose of grounding the reuse and extension of the proposed *upper-level ontology*, the definitions of context information and entities given in Table 13 are compiled as follows:

- **Context information:**
  - *Time.* The definitions of the Time class fundament the semantic of the primitives that should comprise the modules of time involving timeline of past, present and future. In other words, the Time class can be extended, specialized or assembled by means of modules related to time, i.e., sub-ontologies specifying essential concepts of time such as hour, day, etc., needed to reason about the events triggered from the past through the present into the future. Hence, the situational characterization of an entity that is affected by the context information of time should be modelled through the modules and primitives that extend the semantic of such class. For instance, time at which an entity (e.g., client) invokes another entity (e.g., service).
  - *Profile.* The definitions of the Profile class fundament the semantic of the primitives needed to characterize the profile information of the entities involved in an interaction process. For instance, profile of an entity (e.g., client) that invokes another entity (e.g., service) to understand its preferences (a type of context information into the Profile class).
  - *Environment.* The definitions of the Environment class ground its modules to provide the primitives needed to characterize all the environment factors of the entities involved in an interaction process. For instance, the social or environmental conditions (two types of context information into the Environment class) that affect an entity (e.g., client) to invoke another entity (e.g., service).
  - *Role.* The definitions of the Role class fundament its future modules to provide the primitives needed to characterize the functions of the entities involved in an interaction process. For instance, an entity (e.g., service) can be accessed by another entity (e.g., client) based on its assigned role.
  - *States and Status.* In this case, some definitions of the States and Status class are specified in terms of another context information (e.g., time). This means that the future modules and primitives of this class used to characterize the status information of the entities can be related

to the primitives of the modules of the Time class to provide assertions of the status of an entity in the timeline of past, present and future. For instance, the emotional status (a type of context information into the States and Status class) of an entity when (it represents the time) invoking a service.
    - *Location.* The definitions of the Location class fundament its future modules to provide the primitives needed to characterize the abstraction of a physical or logical location of the entities (agents or resources) involved in an interaction process. For instance, location where an entity (e.g., client) invokes another entity (e.g., service).
    - *Activity.* The definitions of the Activity class ground the primitives needed to characterize any specific behavior (actions) of the entities involved in an interaction process. Moreover, this type of context information can be related to another context information (e.g. time and location) to formalize the situational context of an entity through different dimensions that can generate more sophisticated assertions about the activity of an entity. For instance, the activity that an entity (e.g., client) is carrying out at the time it invokes another entity (e.g., service).
- **Entities*:***
    - *Resource.* The definitions of the Resource class fundament the future modules and primitives needed to conceptualize any resource, i.e., through this generic class a modeler can specify different types of entities representing a service, application, tool, etc., that are needed by another entity (e.g., person) to perform something (e.g., activity). As it can be seen, the definition of the Resource class is in terms of a context information (activity), this also means that we can assert that the activity performed by an entity is also affecting, for instance, another context information (e.g., the status of the entity).
    - *Agent.* Contrary to the definitions of the Resource class, the definitions of the Agent class represent the basis to conceptualize different modules and primitives of different agents such as a person, people, a group, an organization, etc. that can act on behalf of other agents or by themselves. Note that the term Agent was selected due to the patterns found in the surveyed models to represent a person, user, client, etc. that generally represented this type of entities as a specialization of an agent. However, due to the generic purpose of the proposed *upper-level ontology*, i.e., it can be also extended at the same level of abstraction. A modeler can take the decision to represent such type of entities out of the scope of the suggested semantic of an agent. Hence, beyond representing two types of entities (Agent and Resource) a modeler can suggest to represent a person or user at the same level of abstraction. This type of modelling decision should not affect the reusability and extensibility of the proposed model while the modeler remains consistent with its proposal and the modelling considerations pointed out in this chapter of the thesis. Such principle apply also for the Resource class and the context information classes.

As it can be seen, the *upper-level* classes described above provide the initial semantic principles and criteria to build the middle and lower levels of the ontology.

3.3.2 Middle-level context ontology

The *middle-level* represents a bridge between the *upper-level* and the *lower-level* of the proposed three-level context ontology and provides a set of ontological modules easy to reuse and extend. To address this modular perspective, we propose reusing a set of ontological resources from existing contributions that can represent structured modules of context knowledge. From this perspective, the proposed context model can be aligned with the most appropriate aspects of existing context ontologies. Hence, the *middle-level ontology* is defined and structured in a modular way for supporting reuse by following the integration process prescribed by Pinto and Martins [218]. The extension and consolidation tasks addressed at this level of the ontology are carried out by deepening into the analysis, selection and combination of many useful vocabularies from different existing proposals. In this regard, the integration process allows identifying and gathering parts of different ontologies to be integrated systematically into modules. From this initiative, we addressed some gaps of a generic context model by allowing the definition and instantiation of new or existing context knowledge in a unique and simple way. The complete integration process is applied below.

3.3.2.1 Identify integration possibility

In this step, we are interested on importing specific modules from existing ontologies in an easy way providing a clear schema of reutilization, and connection with the upper-level classes. To do so, it is necessary to select a tool to support the ontology construction. Following the criteria provided by Su and Ilebrekke [220], we have selected Protégé as ontology development tool. This implies that, in cases where the ontologies selected are provided in a different framework, we will translate the selected modules into the semantics of Protégé. It is clear thus that the selection of this tool is greatly influencing our proposal. This is unavoidable because we want an ontology that can be used in an engineering context in order to provide tangible value in the development of context-aware services.

3.3.2.2 Identify modules in which the ontology can be divided into

We select as modules of the middle-level ontology the context classes established in the upper-level ontology (see Fig. 13). Hence, the subontologies/modules should comprise the context knowledge pieces aligned with the semantic of the high level classes defined in the upper-level ontology, i.e., time, profile, states and status, environment, role, location, activity, resource and agent.

3.3.2.3 Identify assumptions and ontological commitments

To cover this task, we define the assumptions and ontological commitments that each module should comply among them and the resulting model, based on the specification requirements of the three-level ontology as suggested by Pinto and Martins. For this purpose, we follow the guidelines provided in Methontology [58] specifically from the specification phase that allows to produce the specification document that describes the domain, purpose and scope of the future ontology. Such specifications were already stated in Section 3.2 of this chapter. Table 14 summarizes the ontology requirements specification supporting assumptions and commitments of the model.

Table 14. Requirements specification for supporting assumptions and commitments

Domain:	Context aligned with the definition given by Dey [3] and from a service-centric perspective
Purpose:	Ontology about the representation of context knowledge to be used when a unified, consolidated and well-defined context information is required in context-aware services, and for facing different gaps in context modelling.
Scope:	-Representation of context knowledge and entities identified in the review on context modelling conducted in Chapter 2; -Representation of context knowledge causing either positive or negative effects on entities involved in the value structure given by services -Provide a taxonomy of high-level classes to facilitate the extension of the model (upper-level); -Provide modules aligned with the semantic of high-level classes (middle-level); -Provide the initial criteria for grounding domain-ontologies (lower-level)
Sources of knowledge	-Systematic mapping on context modelling conducted in Chapter 2 -Brainstorming with experts on ontologies and services

3.3.2.4 *Identify knowledge to be represented in each module*

For the purpose of this task, we provide a list of essential vocabulary that should aid to compose the modules of the future ontology. As suggested by Pinto and Martins, the conceptual model of the ontology and abstraction capabilities can produce such list. In Table 15 is provided a glossary of context terms that we have gathered from the previous specifications of the ontology, the review on context modelling conducted in Chapter 2, and the brainstorming generated with service experts.

It is worth noting that the list of terms specified in the table is not intended to be a comprehensive list, as Pinto and Martins state it is only to have an idea of what the modules should contain in order to recognize whether available ontologies are adequate to be reused. In other words, the primitives (concepts and properties) of Table 15 represent both repetitive terms found in the study on context modelling and a basis of context knowledge that can fundament each module of the middle-level ontology.

Table 15. Glossary of context terms

Activity module		Location module		Profile module	
Term	***Type***	**Term**	***Type***	**Term**	***Type***
Action	Concept	Indoor space	Concept	Community Prof.	Concept
Deduced activity	Concept	Coordinates	Concept	Device profile	Concept
Event	Concept	Outdoor space	Concept	Location profile	Concept
Process	Concept	Region	Concept	Network profile	Concept
Scheduled activity	Concept	Relative location	Concept	Object profile	Concept
Task	Concept	Building	Concept	Service profile	Concept
changes	Property	contains	Property	User profile	Concept
benefits	Property	coordinates	Property	aim	Property
causes	Property	claims	Property	depiction	Property
hasEvent	Property	existsIn	Property	dislikes	Property
hasPerformance	Property	hasPostCity	Property	account	Property
moves	Property	hasTenant	Property	depicts	Property
Environment module		Role module		Resource module	
Term	***Type***	**Term**	***Type***	**Term**	***Type***
Envir. conditions	Concept	Social role	Concept	Object	Concept
Social envir.	Concept	Civilian	Concept	Comp. entity	Concept
Regulation	Concept	Owner	Concept	Managed entity	Concept
Air pollutant	Concept	Provider	Concept	Proposition	Concept
forbids	Property	User	Concept	Service	Concept
isRegulatedBy	Property	plays	Property	fills	Property
hasAirTemp	Property	hasOwner	Property	consists	Property
States&Status		Time module		Agent module	
Term	***Type***	**Term**	***Type***	**Term**	***Type***
Current status	Concept	Time zone	Concept	Organism	Concept
Cognitive	Concept	Date	Concept	Commercial agent	Concept
Past status	Concept	Relative time	Concept	Group	Concept
Biological state	Concept	Day	Concept	believes	Property
Discrete state	Concept	after	Property	cohabitant	Property
Continuos state	Concept	before	Property	associated	Property
Future status	Concept	day	Property	defines	Property
hasStatus	Property	earlier	Property	desires	Property
isAffectedBy	Property	finishes	Property	employs	Property
isRelatedTo	Property	during	Property	executes	Property

The primitives presented in Table 15 are the basis to search ontologies that better fit and structure at the correct level of abstraction (also fixed under the pattern approach) these primitives. For instance, Air pollutant of the Environment module might be a middle-level class that can be extended and structured by different types of air pollutants (e.g. gas pollutants, radioactive pollutants, etc.) that can also be extended by more specific concepts (e.g., nitrogen oxides *extending* gas pollutants). This modelling consideration suggests that the concepts of a domain-level ontology should be carefully mapped with the concepts of the future middle-level ontology to maintain the consistency among the three levels of the proposed model. The complete structure and formalization of modules are completed through the remaining tasks.

3.3.2.5 Identify and get candidate ontologies

According to Pinto and Martins, this task first identifies candidate ontologies that could be used as modules of the middle-level ontology. We selected 64 context models coming from the systematic mapping reported in Chapter 2 as possible candidates to be integrated in the modules of the middle-level ontology[7]. The selection criteria were: 1) the models provide several concepts and properties as the required in the previous step; 2) the models are based on ontologies; and 3) the ontological resources offered by the models provide context knowledge that matches the modules identified in the middle-level ontology. It is worth noting that the 138 context models (selected in Chapter 2) were also analysed in order to identify possible lack of information in the selected models.

To obtain the candidate ontologies in an adequate form, we analysed their knowledge and implementation levels as well as the documentation available. At this respect, although the knowledge level was found in most of the selected ontologies, generally it was not deeply detailed. Similarly, the implementation level of some of the ontologies was only partially defined and their availability in ontology libraries was almost inexistent. However, for each model we aimed at identifying and retrieving the ontological resources that we considered relevant to create or complement modules of the middle-level. The relevance has to do both with aspects such as frequent classes, common patterns identified through the proposed schemas, etc., and the considerations of the previous tasks to identify essential concepts. It is worth to mention that we considered not only the 64 context ontologies but also other 12 ontologies that were reused by them. We decided to establish a common base of candidate ontologies acting as reference point to structure the modules required. To carry out this task, from these ontologies, we selected those ones more referenced by existing proposals of context modelling considering their adequacy for representing knowledge of future modules as previously specified: CONON [19], SOUPA [17], SUMO [76], OpenCyc [62], FOAF[8], CCPP[9], OWL-Time[10] and OWL-S[11]. The remainder tasks of the integration process are focused on them.

3.3.2.6 Study and analysis of candidate ontologies

In this task, the candidate ontologies are analysed to identify possible problems in the integration process. We applied the SEQUAL evaluation framework formulated by Hella and Krogstie [221], which has been used for similar evaluations in a number of related areas such as goal models, requirement models, etc. In such cases, SEQUAL has been used by specializing the generic framework to the relevant domain and goal of modelling. There are 7 quality categories used to evaluate the reusability of the ontologies (see below). For each category, Hella and Krogstie propose some values that we have mostly kept, except for a couple of minor changes: 1) we added values that were not specified to describe some features of the current status of the ontologies, for instance we added the value “opens with certain problems” to specify

[7] The 64 references corresponding to the selected ontologies can be found at [**Error! Reference source not found.**]
[8] xmlns.com/foaf/spec
[9] www.w3.org/TR/CCPP-struct-vocab2/
[10] www.w3.org/TR/owl-time
[11] www.w3.org/Submission/OWL-S

some problems found when open an ontology as the too much spent time to charge the ontology, sometimes the complete ontology was not opened, etc.; 2) we added "RDF" as syntactic format; 3) we changed the "OK" values by the term "satisfactory". The results of the evaluation are depicted in Table 16 and the categories are described as follows:

- *Physical (Phy)*. The ontology should be physically available and it should be possible to make changes to it. Available (✓); available, presenting some problems to open in Protégé (✓-); available, but too big to open (✓--); not available (X).
- *Empirical (Emp)*. If a visual representation of the ontology is provided it should be intuitively and easy to understand. Satisfactory (✓); less satisfactory (✓-).
- *Syntactic (Syn)*. The ontology should be represented according to the syntax of a preferred machine readable language. OWL full (✓); partial OWL (✓-); RDF (✓--).
- *Semantic (Sem)*. The ontology should cover the area of interest. Overlap, satisfactory validity (✓✓); partial overlap but not complete, satisfactory validity (-✓); partial overlap but not complete, poor validity (--); not overlapping (X). Since this category is too coarse to be applied globally, Table 17 shows its evaluation for the modules identified in the middle-level ontology.
- *Pragmatic (Prag)*. It should be possible to understand what the ontology contains, and being able to use it for our purpose. Satisfactory (✓); not satisfactory (X).
- *Social (Soc)*. The ontology should have a relatively large group of users. Mature and widely used (✓✓); assumed mature, not specified how much it is used (--); not mature, but referenced (-✓).
- *Organizational (Org)*. The ontology should be freely available, accessible, maintained and supported. Free, accessible, and stable (✓✓✓); free, accessible, and probably stable (✓✓-); free, not accessible, and probably stable (✓x-).

Hella and Krogstie already provided in [221] the evaluation of some of the candidate ontologies selected in the previous task, concretely FOAF, OpenCyc and SUMO. We reused this evaluation reviewing the current status of the ontologies in order to check that the results obtained remain consistent; the only change has already been reported above (currently SUMO can be opened in Protégé). The rest of ontologies were evaluated from scratch. The results obtained are depicted in Table 16 and Table 17.

Table 16. Evaluation of candidate ontologies

Ontologies	*Phy*	*Emp*	*Syn*	*Prag*	*Soc*	*Org*
CONON	X	✓	✓-	✓	-✓	✓x-
SOUPA	✓	✓-	✓-	✓	-✓	✓✓-
SUMO	✓-	✓	✓	X	--	✓✓-
OpenCyc	✓--	✓-	✓	X	--	✓✓-
FOAF	✓	✓	✓	✓	✓✓	✓✓✓
OWL-Time	✓	✓-	✓	✓	--	✓✓-
CCPP	✓	✓	✓--	✓	--	✓✓-
OWL-S	✓	✓-	✓	✓	--	✓✓-

Table 17. Semantic evaluation of candidate ontologies organized by modules

Ontologies	*Agent*	*Resource*	*Activity*	*Time*	*Environment*	*Location*	*Profile*	*Role*	*Status*
CONON	-✓	-✓	-✓	X	X	-✓	--	X	X
SOUPA	-✓	--	-✓	✓✓	X	-✓	-✓	--	--
SUMO	-✓	-✓	-✓	✓✓	--	-✓	--	--	--
OpenCyc	--	--	--	--	--	--	--	--	--
FOAF	X	X	X	X	X	X	-✓	X	X
OWL-Time	X	X	X	✓✓	X	X	X	X	X
CCPP	X	X	X	X	X	X	-✓	X	X
OWL-S	X	X	X	X	X	X	-✓	X	X

3.3.2.7 Choosing source ontologies

Given the study and analysis of candidate ontologies the final choices must be made in this task. Pinto and Martins propose two stages. In a first stage, a critical look to the characteristics analyzed in the previous task is made[12].

Although the schema presented by SOUPA and CONON for context modelling is widely referenced in the academic research, the major drawback of both ontologies is the resources provided: they are not fully available or cannot be imported into Protégé, they are not maintained and some context information and entities are not considered. Semantically, both SOUPA and CONON are small ontologies with few classes and relationships that partially characterize the situation of a few entities. Still, the design of the model presented here is partially inspired by the modular schema of SOUPA and the intuitive visual representation of CONON.

The rest of ontologies are computationally available. However, SUMO and OpenCyc are big ontologies difficult to import into ontology editors (this fact was already reported in [221]). In fact, the loading time of the OpenCyc ontology into the editors is too long due to its size. Empirically, SUMO, FOAF and CCPP provide a visual representation of their schema that is easy to understand. On the contrary, OpenCyc, OWL-Time and OWL-S do not present such an intuitive schema.

In the semantic and pragmatic qualities, SUMO and OpenCyc are upper ontologies providing an extensive vocabulary; although these vocabularies can be used for purposes of context modelling, a large subset of this vocabulary is irrelevant for this purpose. The rest of the foundational ontologies are more concrete and provide a smaller set of vocabulary partially covering the context of an entity. For instance, FOAF provides simple vocabulary for describing people, what they do and their relationships to other people; OWL-Time provides a simple vocabulary for describing temporal content of web pages and temporal properties of web services.

Based on this assessment, the final decision is made in a second stage. Our aim is to select the parts of each candidate ontology that cover satisfactorily a module identified from the upper-level ontology; also, we consider the overall ontology evaluation to decide whether to include it in the result or not. The result is shown in Fig. 15. As it can be seen, in the middle-level of the model we propose different modules associated to the corresponding high-level classes of the upper-level ontology. These modules are selected from the candidate ontologies by means of the following considerations: 1) integrate modules fulfilling the conceptualization of a given entity or context information; 2) otherwise, a new module combining ontological resources from different sources is proposed. To support these considerations, we appraised the requirements specification stated in Section 3.2 and Table 14, the glossary of essential terms specified in Table 15 and the evaluation provided in Table 16 and Table 17. We also illustrate how lower-level, domain-specific ontologies are related to middle-level modules (see Section 3.3.3 and 3.4.1 for details).

Several situations have been found when selecting the modules. For instance, the *Object* module is selected from SUMO since it provides the overlap required to conceptualize this module. However, this ontology does not provide at all the required resources to conceptualize a computational entity, so we complement it with resources from CONON. Another case is presented in the *Environment* and *Role* classes from which we have not found in the candidate ontologies the vocabulary that clearly overlap certain modules such as regulation, environmental conditions, roles, etc. In this case, we aim at selecting and combining vocabulary from all the ontologies evaluated to define new modules matching the semantic required by the parent classes.

[12] Although Pinto and Martins use different criteria for evaluation, we find more natural to base the selection on the analysis previously made.

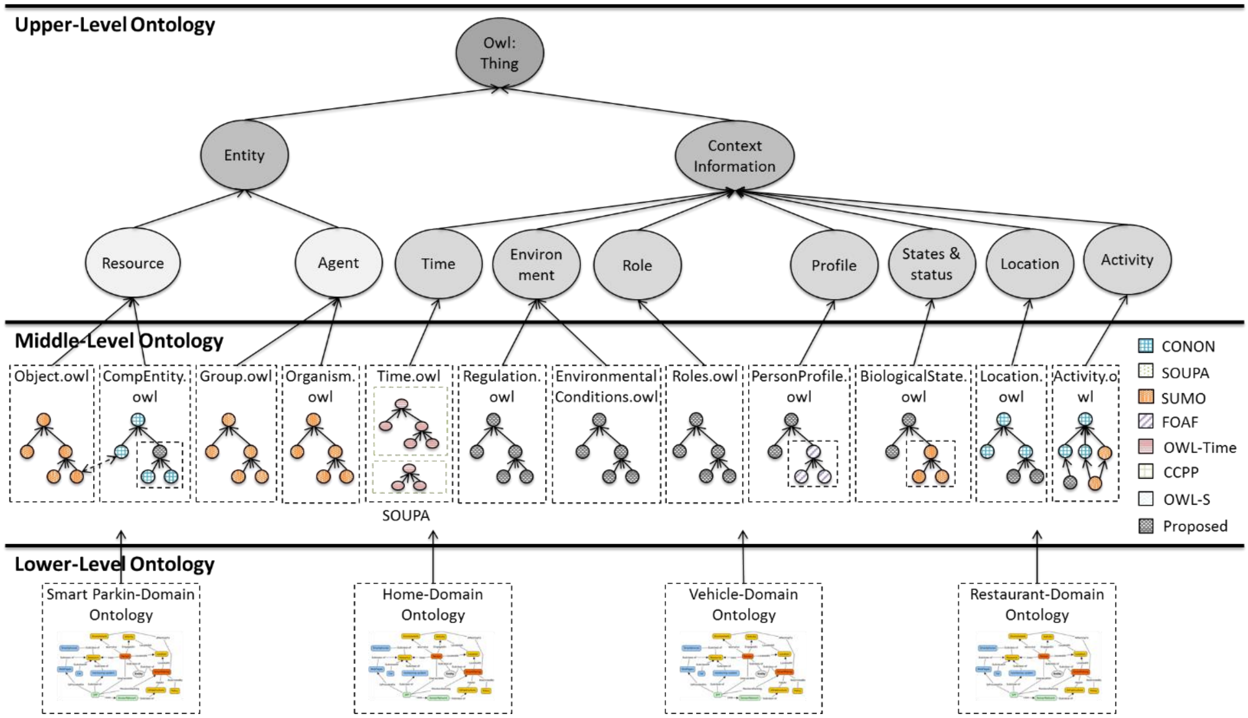


Fig. 15 Middle-level ontology and its relationships with the upper and lower levels

A third case occurs in the *Profile* class; although the FOAF ontology overlaps the semantics required to conceptualize a person's profile, it does not provide vocabulary to describe an object, a computational entity, etc. Specific details about the vocabulary used and combined are given in the next step where the integration task is performed.

3.3.2.8 Apply integration operations

Once the candidate ontologies have been filtered, the final task is to perform their integration (see Fig. 16).

Consider the following case. The upper-level class *Time* can be structured by using the semantics of SUMO, SOUPA or OWL-Time as it is depicted in the semantic evaluation of Table 17. However, according to the evaluation given in Table 16, it is difficult to identify certain resources from SUMO. In this regard, we took as a basis the semantics of time given by OWL-Time (formerly DAML-Time) because it is particularly focused on modelling time and for other features also evaluated in Table 16 (e.g., it is detailed and well-documented). Then, we adopted the following integration operations to provide the most suitable module of time: 1) we integrated in a module the OWL-Time as it is and then we make some modifications in the structure and vocabulary taking into account the next operation; 2) we identified the equivalent resources among the vocabulary and patterns presented in ontologies assessed where time was also modelled in order to unify, consolidate and minimize semantic inconsistencies among these ontologies and OWL-Time ontology (see Fig. 16.A).

Consider a second case. The semantics of the *Location* class at the upper-level ontology has been matched with the conceptualization and structure of the *Location* class given in CONON; however, as shown in Table 16, this ontology provides a smaller set of vocabulary. Hence, we performed the knowledge augmented integration operation by semantic completeness, i.e., we retrieved vocabulary regarding location from candidate and consensus ontologies, integrating them following the semantic of CONON to minimize inconsistences (e.g., we integrated the *GeographicalPlace* class that is highly considered in the vocabulary of location as a subclass of *OutdoorSpace* specified in CONON). We also augmented knowledge considering the definition of all the classes involved in a module (e.g., the *RelativeLocation* class extent the structure of *Location* given in CONON and we also specified *Indoor* and *Outdoor* classes as a subclasses of this class as depicted in Fig. 16.B).

As a third case, modules to structure the *States and Status* class of the upper-level ontology were not found, i.e., the candidate and consensus ontologies do not provide a well-defined structure to conceptualize different vocabulary regarding states and status. At this respect, we retrieved from the mentioned ontologies all the vocabulary that defines a pattern and that overlaps with the semantics of this high-level class. For instance, we built from scratch the module of *BiologicalState* extending our structure by augmenting modules from other ontologies (e.g., SUMO provides a good overlapping with the *StateOfMind* as depicted in Fig. 16.C). Similarly, modules to structure the *Environment*, *Role* and *Profile* classes of the upper-level ontology were not found. Hence, we also retrieve from the studied ontologies all the primitives aligned with patterns and semantics of these high-level classes (see Fig. 16.F.G.H). Note that in the case of the Profile class we augmented the knowledge of the *PersonProfile* module by considering the good overlapping of the FOAF ontology to describe a person's profile.

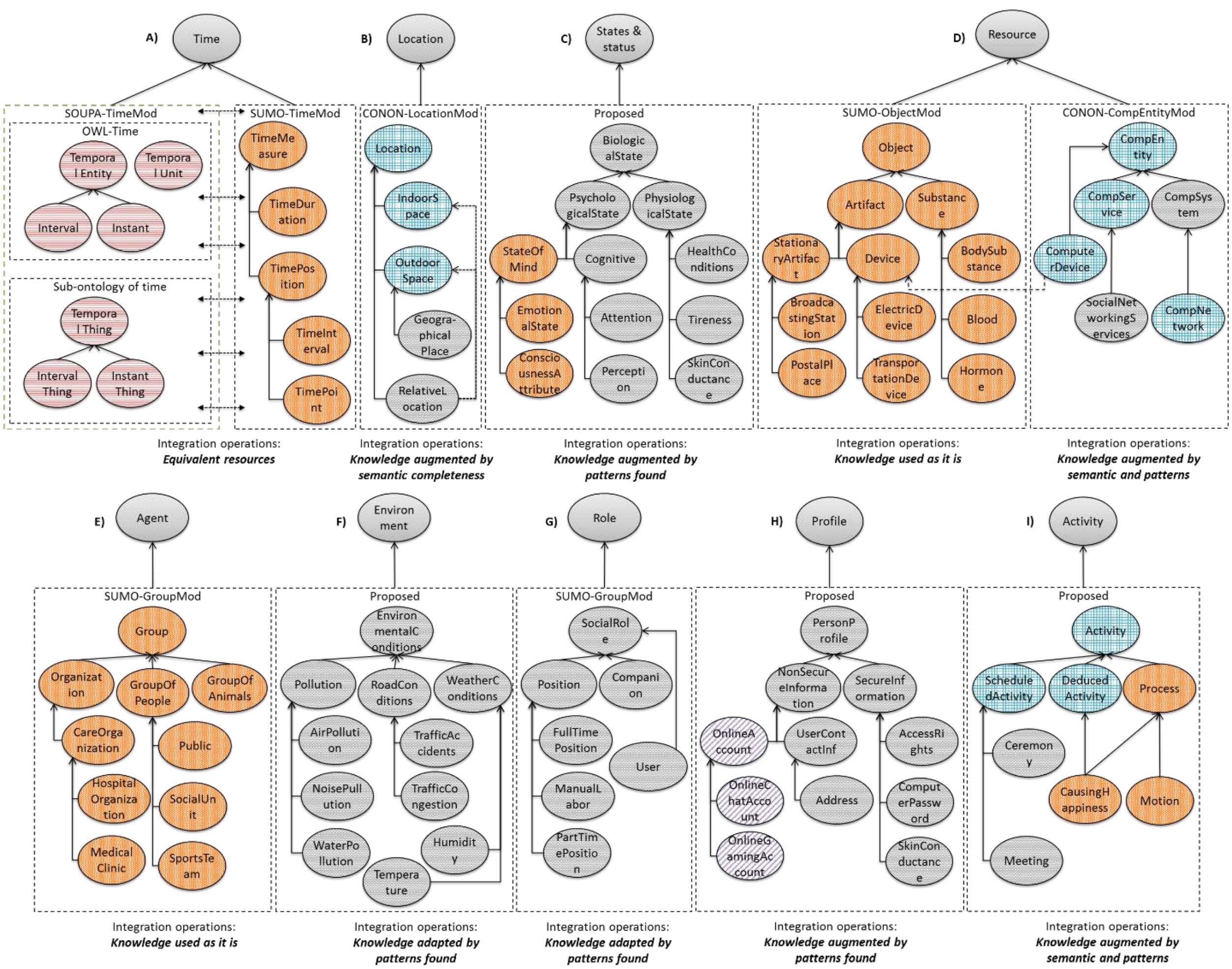


Fig. 16 Detail of the middle-level ontology

In a four case, we refer to the *Resource*, *Agent* and *Activity* classes of the upper-level ontology (see Fig. 16.D.E.I). As it can be seen, in the Resource class we have 1) used as it is, the structure and knowledge given by the SUMO ontology to define the *Object* module since it provides a good overlapping with the semantic required and presented in the patterns of different proposals of context modelling; 2) augmented by semantic the knowledge of the *ComputationalEntity* module taking as a basis the CONON ontology and restructured by patterns the knowledge specified; and 3) search equivalences and generic types among the classes represented in the modules of the *Resource* class to decrease inconsistencies. Similarly, this process is applied to the modules of the *Agent* and *Activity* upper-level classes.

To sum up, we highlight the benefits of the proposed middle-level ontology for context modelling once the integration process is applied. Firstly, it is important to remark the restructuration and renovation of several ontological resources of different ontologies that we have reused in our model given the patterns found in existing contributions. For instance, we found that several data and object properties defined in different models are outdated and therefore, they are not used, however these archaic resources are still provided (e.g., FOAF and OWL-time provide properties with this feature). In the same way, documentation and implementation of the model are also outdated and therefore they are not aligned between them.

In the OWL-time ontology we identified that several datatypes are not supported by most of the reasoners currently available (e.g., gday datatype). In fact, only the Pellet[13] reasoner supports most of the datatypes provided in this ontology. In relation to this issue, we have updated and tested the usage of different ontological resources by means of their instantiation in different usage scenarios (see Section 3.4.1). We also have faced the unification, consolidation and semantics of the ontological resources in order to avoid inconsistencies among them by analyzing their definitions and usage in different proposals (e.g., “interest” that is a property in FOAF was stated as “hasInterestOn” since most of the existing proposals use this term to refer the interest of someone about something; similarly, “topic” property was renamed as “hasTopic”; and so on). Finally, in order to avoid wrong definitions in the ontology and its associated modules, we carried out an evaluation during and until the end of the development of them by inspecting the ontology-based categorization to avoid possible errors as recommended by Gómez-Pérez [223].

3.3.3 Lower-level context ontology

The lower-level ontology of the 3LConOnt context model represents domain-specific ontologies whose vocabulary, i.e. classes, properties (object, data and notation), etc., is highly dependent on the domain. According to Kishore and Sharman, lower-level ontologies pertain to bounded universe of discourses and are referred to in the literature as application, domain and task ontologies [224]. In the three-level approach, lower-level ontologies developed by the ontology designer are grounded on the upper and middle level ontologies, i.e., lower-level ontologies are created following a set of initial criteria and semantic principles given by the middle and upper levels of the model.

In Fig. 17, we depict a lower-level ontology designed from the upper and middle-level ones. This ontology has the aim of conceptualizing the input, output and general capabilities of monitoring tools. As depicted in the figure, middle-level classes have been extended by lower-level classes (domain classes). From this perspective, we can conceptualize different context information (e.g. time, social environment, location, etc.) and entities (e.g. applications, monitors, feedback mechanisms, etc.) that can be interrelated for conducting different activities (e.g. configure a monitoring tool given certain change in the context information).

As depicted in Fig. 17, a monitoring tool has been conceptualized to represent that “any monitoring tool is a monitoring program which is in turn a computer program, a software and a computational entity”. Such modelling can be applied to any monitoring tool for representing its inputs, outputs and capabilities. For

[13] Pellet is an OWL-reasoner written in Java and provided as open source software supporting SWRL language to describe first order query rules [222].

instance, a feedback gathering tool can be modelled in the same way. It should extend the software class with the class or classes that can represent a feedback gathering tool class that can contain all the feedback mechanisms related. Following the conceptualization of the domain ontology, we are able to represent that "any instance of the SocialNetworksMonitoring class is a monitoring tool focused on monitoring different social network services such as Facebook and Twitter that have comments of people belonging to a social environment". For a generic description, we are able to represent that any instance of the SocialNetworksMonitoring class is related to only one instance of the SocialNetworksMonitoringP-rofile class for describing its profile information such as extended name, the URL prefix representing the URI for triggering requests to the server and more detailed description.

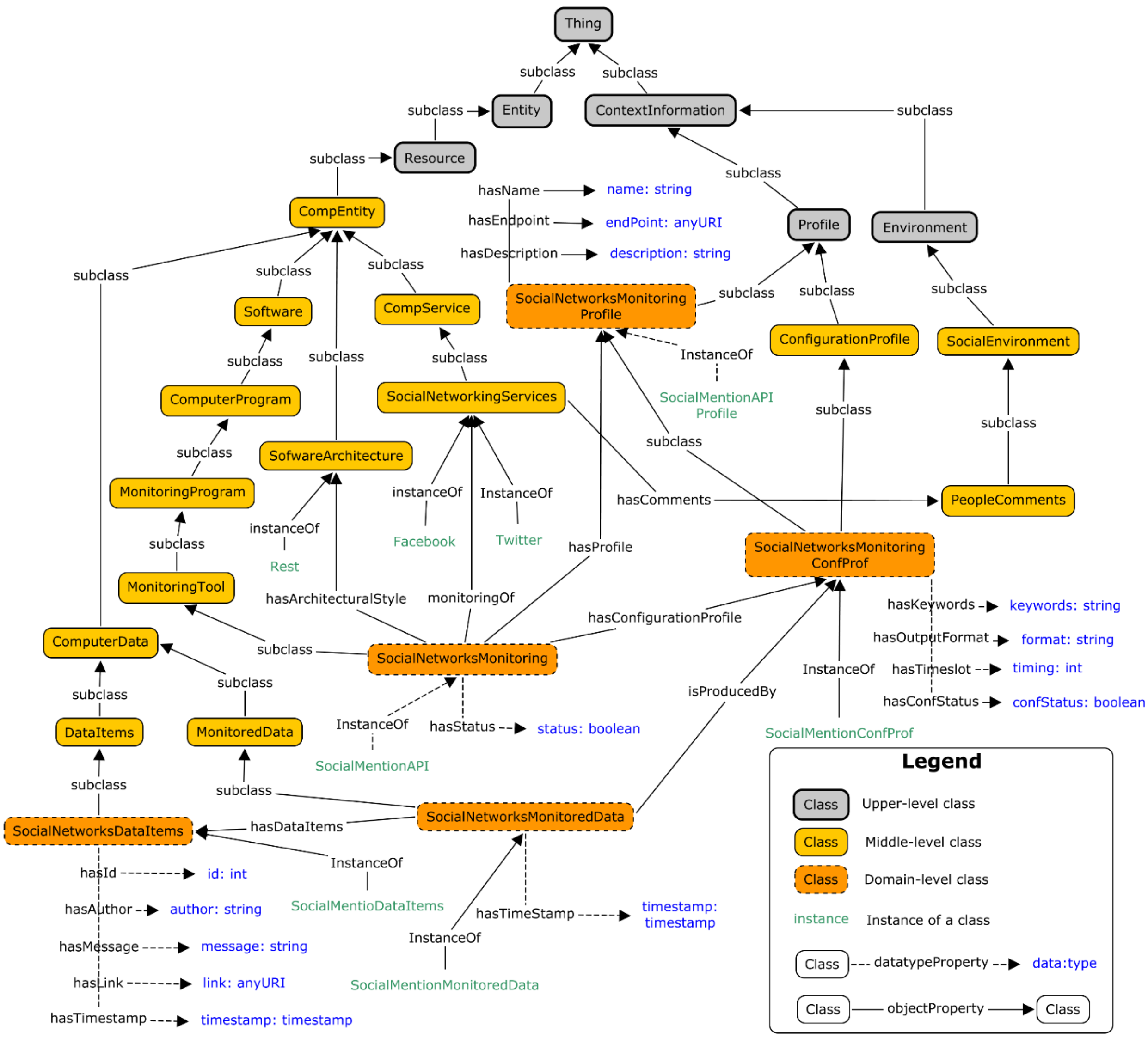


Fig. 17 Ontology for monitoring conceptualization at three-levels of abstraction

The input of a monitoring tool is conceptualized to represent that any instance of the SocialNetworksMonitoring class has a Boolean status On or Off indicating if the monitor is activated or not. Each instance of the SocialNetworksMonitoring class can be related to one or more instances of the SocialNetworksMonitoringConfProf class consisting of different parameters that can be configured such as the keywords that are going to be searched, the response format (json, php, xml, rss, csv), etc. Finally, the output of a monitoring tool (data monitored) is modelled for indicating that "one or more instances of the SocialNetworksMonitoredData class is produced by an instance of the SocialNetworksMonitoringConfProf class. Each instance of the SocialNetworksMonitoredData class has a timestamp and one or more number of data items (instances of the SocialNetworksDataItems class) each of them consisting of different response properties such as id (unique hash id), message, link, timestamp, author, etc.”.

The benefits of the proposed lower-level ontology include the provisioning of a unified and representative schema of monitored data, as well as the provisioning of a clear schema of inputs and outputs of monitoring tools. In this regard, we intent to response some competency questions with the aim of evaluating that the model provides the set of axioms to represent and solve such questions. Hence, some of the competency questions that can be answered in the proposed domain ontology are the following:

- What are the parameters that can be configured in a monitoring tool?
- What kind of context information can be monitored by a monitoring tool?
- What are the monitored data that are related to a specific parameter (e.g. retrieve all the monitored data that are related to certain date)?
- What kind of response format is given by certain configuration instance?
- What instances of a monitoring tool are activated?
- What are the monitoring tools that can monitor a specific service or application (e.g. retrieve all the monitors that can monitor the Twitter)?

In summary, the upper and middle levels of the ontology intent to capture and model basic concepts and knowledge of context that can be reused by the ontology designer for building new domain-specific ontologies (lower-level ontologies) or for reusing existing ones following the same criteria. From this perspective, different domain specific ontologies can be developed and integrated in the proposed model to represent common concepts and relations that are typical in a particular topic. For more reference of lower-level ontologies, we provide a set of concrete scenarios in Section 3.4.1 that validate the generality of the proposed context model.

3.3.3.1 Supporting material for building domain ontologies

In Fig. 17, we conceptualized both the abstract layer (including upper and middle level classes) and the domain layer (specifying domain-level classes) of the ontology for conceptualizing monitoring tools. For the domain layer, we provide as an example a new type of monitoring tool named *SocialNetworksMonitoring* that may help the reader to analogously create a new type of monitoring tool. We describe in Table 18 to Table 22, the input and output data needed by each domain-level class depicted in Fig. 17. This information (hierarchy and glossary of terms) simulate the state that a modeler interested in using this ontology would find as starting point.

Table 18. Glossary of data – SocialNetworksMonitoring

Parameter/data	**description**
idSNMTool	Identifier of the monitoring tool
statusSNMTool	Status specifying if the monitor is up and running or not

Table 19. Glossary of data – SocialNetworksMonitoringProfile

Parameter/data	**description**
Name	Name of the monitoring tool
endpointSNMTool	URI where the monitoring tool is located and can be accessed
description	A short description of the monitoring tool

Table 20. Glossary of data – SocialNetworksMonitoringConfProf

Parameter/data	**description**
idConf	Identifier of a configuration
confStatus	Status specifying if the configuration is up and running or not
Accounts	Accounts that are going to be monitored
Keywords	The keywords that we want to search for
Timeslot	The timeslot between two monitoring invocations
kafkatopic	The topic that is used for storing the monitored data in Kafka (a messaging system to capture and publish data).

Table 21. Glossary of data – SocialNetworksMonitoredData

Parameter/data	**description**
idOutput	Identifier of each output of the monitoring tool
searchTimestamp	The timestamp in which the data has been collected
numDataItems	How much data items have been collected

Table 22. Glossary of data – SocialNetworksDataItems

Parameter/data	**description**
idItem	The identifier of the item
Author	The author of the social network message
Message	The text of a social network message
Link	The link to the original message
timestamp	The timestamp in which that message was created

3.4 Validation of the proposal

In this section we validate the proposal considering: 1) its level of generality validating its reusability, extensibility and adaptation through different smart scenarios; 2) its consistency and reasoning by triggering queries to the proposed model based on some competence questions and from the perspective of a smart restaurant scenario; 3) its reusability in existing context ontologies demonstrating that each level or module of the model is independent of the entire model; and 4) its correctness and completeness for demonstrating that the ontology is consistent and copes the vocabulary needed in the modules.

3.4.1 Evaluating level of generality: use case scenarios

For evaluating the level of generality and consolidation of the proposed model, we analysed if some of the scenarios specified in the literature selected in the systematic mapping (see Chapter 2), are representable using the ontology. Such scenarios have been selected based on: 1) most referenced papers, 2) describing a kind of smart service, and 3) involving different entities and context information.

Scenario 1 – Smart parking scenario. *"Chris is a 50-year old business man that needs to attend an event in the centre of Barcelona and he is thinking of using his car to reach there, but he is worried for the availability of parking spots and the route to arrive there because it is a very busy place. One of his friends advise him to use an application called Smart Parking Platform (SPP) that has different capabilities that improve the user experience, such as checking and booking free spaces in the area; searching simple route (basic use) or searching and being aware of other context e.g., strikes, accidents, etc. to provide the best route (advanced use); adapting or evolving its behaviour and interface since it is also aware of the user experience in using the app; etc. SPP uses a sensor network that continuously* *monitors parking spot activity and incoming reservation requests, based on this information SPP website shows a map with lights indicators, green (unoccupied), yellow (reserved) and red (occupied).*

Once Chris has booked parking through SPP website the system updates the map with lights indicators from green to yellow. Then, just before entering the garage, a system in place requires ID number of reservation request which was sent through SPP website to his smartphone. After validating the ID, the system updates the light indicator on the spot from yellow to red. Once the event has finished, Chris tries returning to the spot where his car is parked, but there is a difficulty, he does not remember the accurate spot, for this, Chris uses a SPP mobile application that from his current location and by means of Google Street view Chris is guided to the parking spot. Finally, when Chris is gone, the system switched the light back to green and charged a parking fee to the driver's account" (see Fig. 18)

As it can be seen, the above scenario involves different primitives (classes of context information and entities, and objects and datatype properties). The lower-level ontology that models these primitives is illustrated in Fig. 18. As shown in this figure, the high-level classes of the upper-level ontology have been redefined by means of the classes of the middle-level ontology yielding the possibility to characterize domain-specific primitives of the Smart Parking scenario. For instance, the SSP class was possible to be characterized through the *MonitoringSystem* class which in turn is subclass of different middle-level classes belonging to the *Resource* class of the upper-level ontology. Furthermore, the individual *Chris* is an instance of the *Human* class that is synonym of person and that can be associated with different instances of other classes of the middle-level ontology to describe the situation of this specific entity.

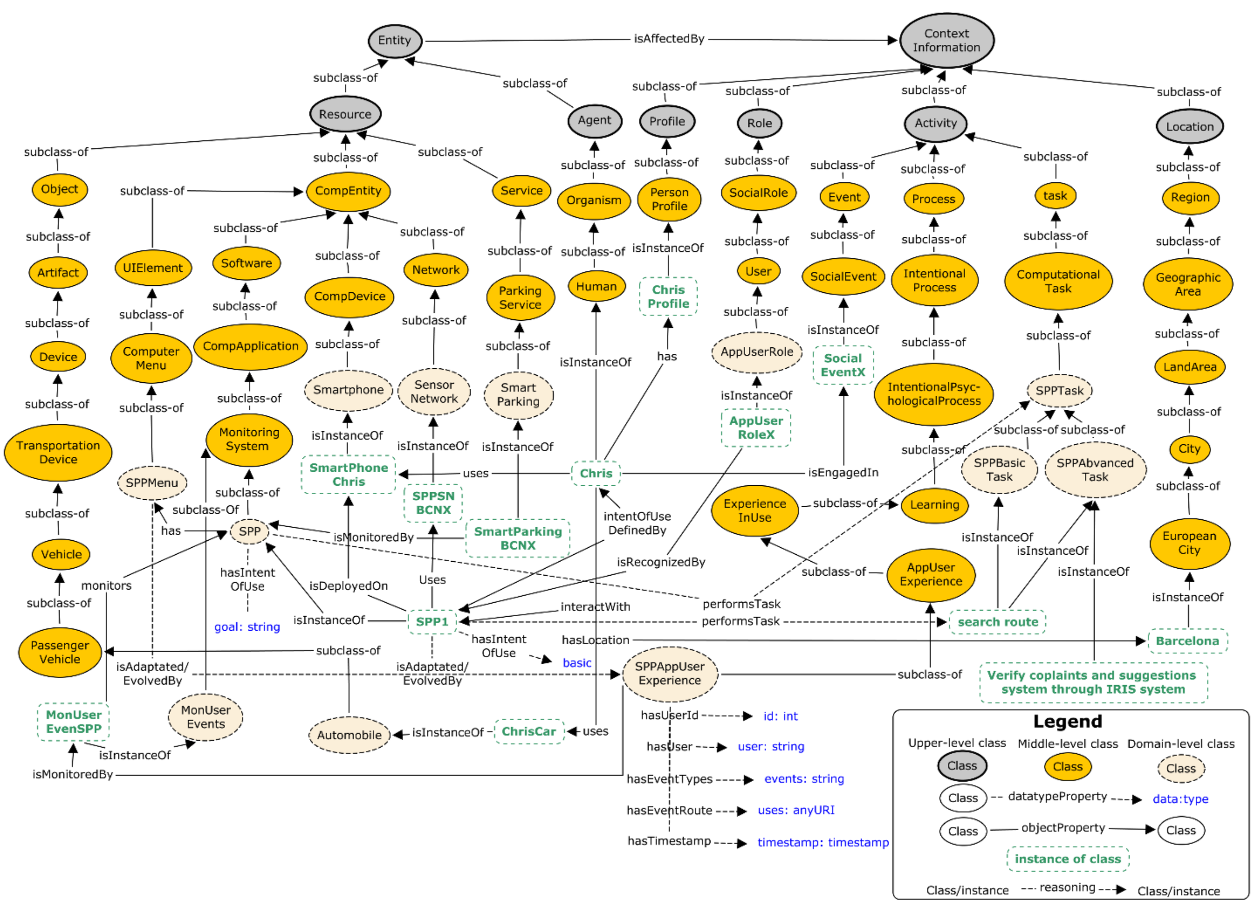


Fig. 18 Smart parking – scenario 1

Regarding reasoning, the ontology can make inferences about the user experience or intention of use to trigger adaptations or evolutions in the behaviour, interface (e.g., menu) or tasks of the SPP. For instance, through reasoning we can specify: if the intention of use of an instance of SPP is defined as *basic* by a user when interacting with such instance, then this instance should be adapted to fulfil some specific tasks. Note that the three levels of the proposed model are illustrated in each ontology representation of the scenarios. Note also that mainly the three-level classes remain static (defined at design time) and their instances can be created and evaluated dynamically (maybe at run time) in a process (e.g., adapt or evolve an application). The three-level classes are used to provide structure, semantic, meaning, consistency, etc. among the primitives involved in a scenario. This static view of the ontology is always important to provide such properties when creating an instance or even if a new class is also dynamically created, i.e., the meaning of such instance or class should be stated to increase the quality, interoperability, etc. of the entire model.

Scenario 2 - Prosumer scenario. "*A particular individual has arrived in a city for the first time and that is travelling along with his wife. That information can be obtained from the location of the mobile devices and querying the context history database. Both devices have been located in the same bearings at the same time (the system concludes through reasoning that these two individuals are located together in the same place)…"* [73] (see Fig. 19).

As it can be seen in the ontology representation of the above scenario (see Fig. 19), there are two domain classes (ProsumerPerson and MobileCellPhone) that were needed to represent some instances of the scenario, and other instances such as CityX and 18:00 were represented directly in middle-level classes. With this representation, we are able to deduce through the location of the mobiles belonging to the individual's X and Y their location and therefore, if they are together in a same place.

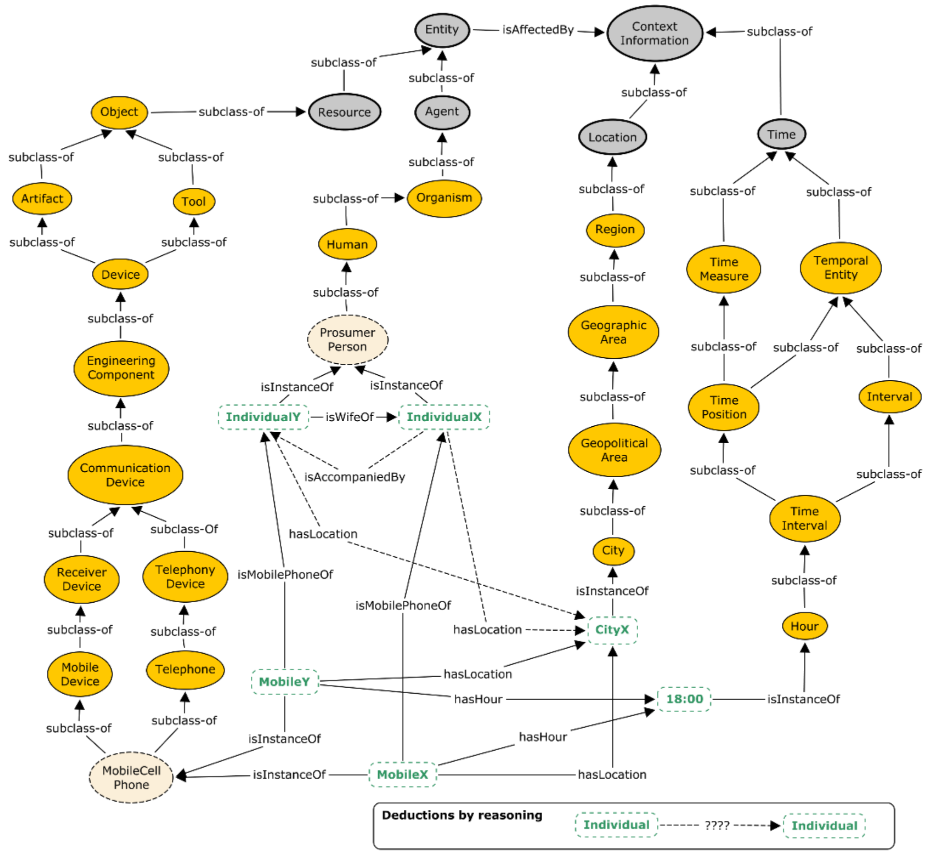


Fig. 19 Prosumer ontology - scenario 2

Scenario 3 – Meeting scenario. "*John Pappas is an executive of an international construction company, who works at the company department that resides in Athens. John is informed that he will have to attend a meeting in Paris for a project he is currently involved in, so he activates his electronic agenda entering the meeting date, place and scope to check his availability…"* [82] (see Fig. 20).

As it can be seen in the ontology representation of the above scenario (see Fig. 20), there are five domain classes such as ElectronicAgenda, Constructuction company, etc., that were needed to represent some instances of the scenario, and similarly to the previous scenario, other instances such as JohnPappas and ProjectX were represented directly in middle-level classes. With this representation, we are able to represent that a ProjectXMeeting has a profile ProjectXMeetingProfile with the following information date, hour and place. Hence, different instances of FormalMeeting can be described through a profile and at the same time, an electronic agenda can manage such meetings.

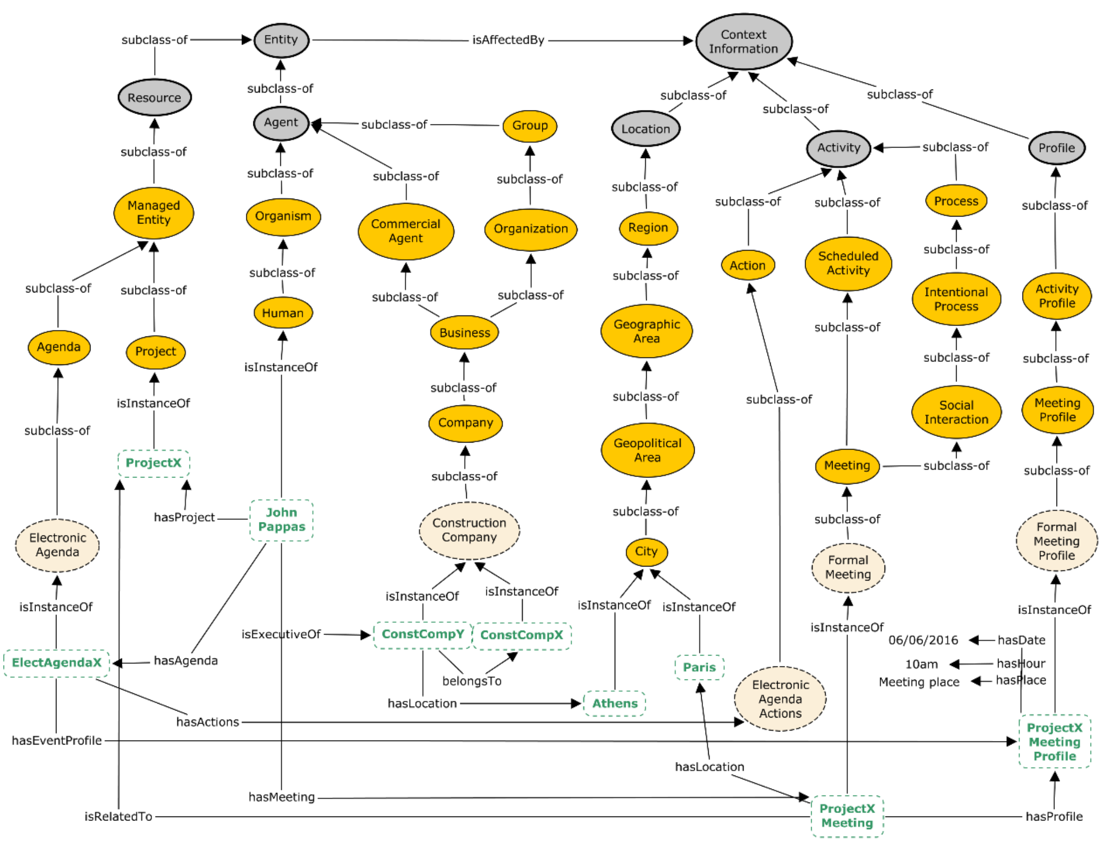


Fig. 20 Meeting ontology – scenario 3

Scenario 4 - Getting Up scenario. "*Mr. Kim sets the getting up time at 6:00 am, and goes to bed late. He must go to his office early. The getting up application checks Mr. Kim's getting up time, and provides an alarm service at 6:00 am the next morning. Then the application opens the curtain to provide fresh morning air and sunshine. The application connects to a weather network service and receives local weather forecast. If it is not rainy, the application system opens the windows. If it is rainy, the system activates the air cleaning service and then provides a light service to supply enough brightness. This getting up application checks Mr. Kim's schedule, and displays it on an output device near Mr. Kim. The system also turns on a display device to show Mr. Kim a morning TV news program. The program information is referred from the preference list which stores Mr. Kim's favorite TV program list."* [84] (see Fig. 21).

Similarly to the previous scenarios, domain-specific entities and context information playing a role in the above scenario were represented through the concepts and primitives of the proposed upper and middle-level ontology. Although we are not pretending to represent extend and exhaustive scenarios with several variables that take part in a context decision, because the aim of the proposal is to provide a coherent model easily reusable and extensible, we considered the full version of the previous scenario as an example of completeness.

As it can be seen, the model depicted in Fig. 21 represents different important aspects of the scenario such as how a window, air cleaning device and light system of a room can be automated and controlled through deductions by reasoning. For instance, if the getting up app detects air pollution at given time and location inferred by the latitude and longitude position of Kim and by the access to the environmental sensors of such location, then some resources of the Kim's room are turned on and others turned off or calibrated. In this case, the upper-level classes needed to provide the formal structure of the scenario are Resource, Agent, Time, Activity, Profile, Location and Environment. Note that some domain classes illustrated in Fig. 21 can be represented directly as middle-level classes, however it is a modelling decision taken by the modeller of the domain ontology. Specially, when a needed class is not yet represented in the modules of the proposed model.

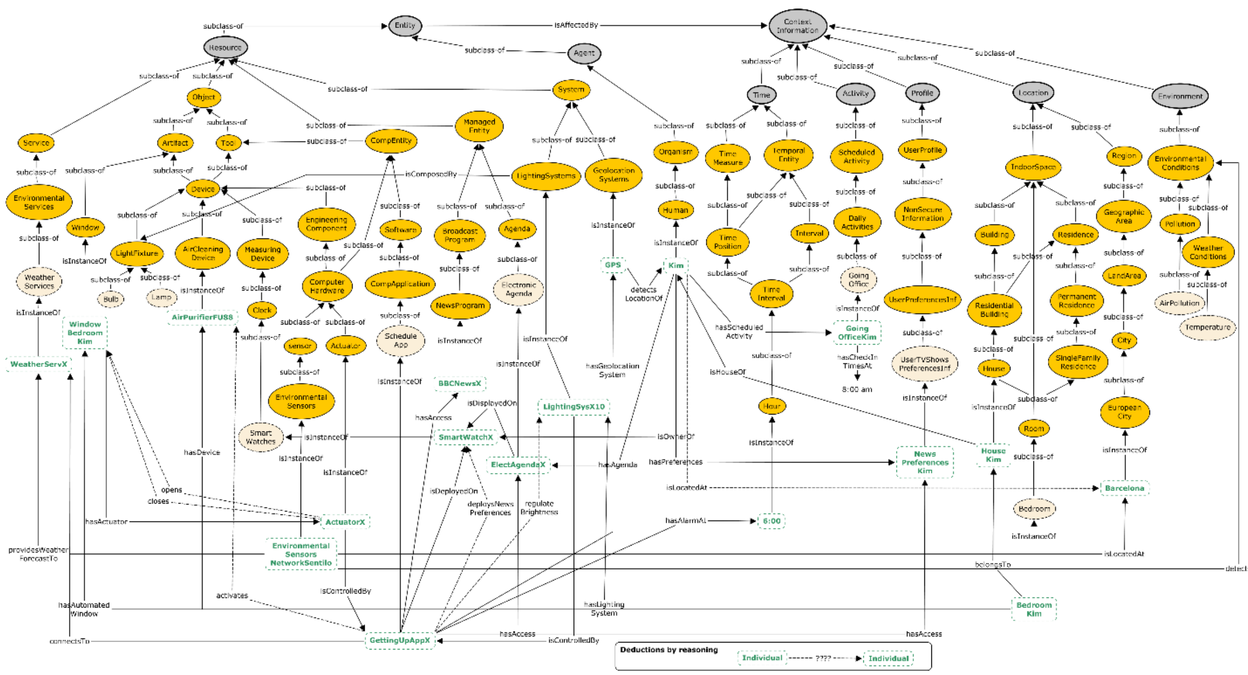


Fig. 21 Getting up ontology – scenario 4

Scenario 5 - DAIDALOS scenario. "*…As soon as Bart is in sufficient range, his RFID unlocks the car and sets the car conditions to suit Bart. When he starts the car, the newscast session he was watching at home resumes in audio only mode at his in-car multi-media system…*" [188] (see Fig. 22).

The instances of the above scenario were represented through middle and domain classes for automating some tasks of a smart car. In this case, the upper-level classes needed to provide the formal structure of the scenario are Resource, Agent, Location and Profile. In general, with these upper-level classes and their corresponding modules of the middle-level classes we are able to represent a multimedia system that is configured by a RFID belonging to specific persons that are owner of a car with a multimedia system.

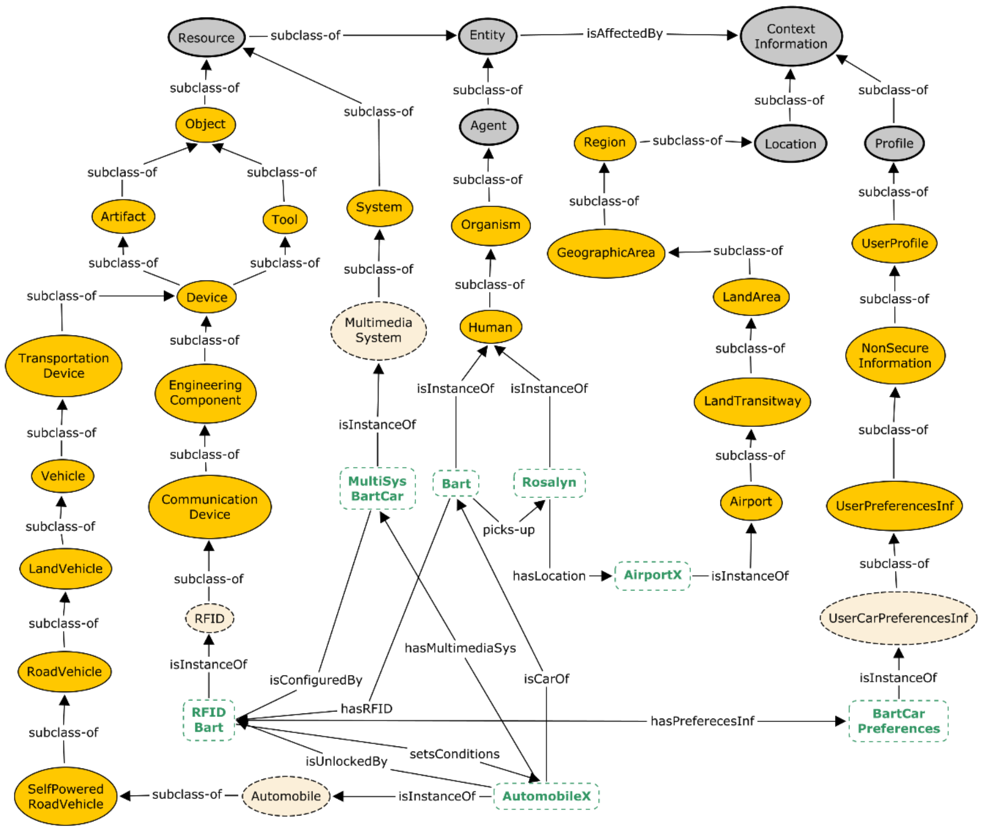


Fig. 22 DAIDALOS ontology – scenario 5

Scenario 6 - An online collaboration service scenario. "*The agent in context tag in the library checks his context and fetches some useful contexts e.g. language preferable (English), his module name, and appreciate communication tool (Messenger) etc…*" [196][225] (see Fig. 23).

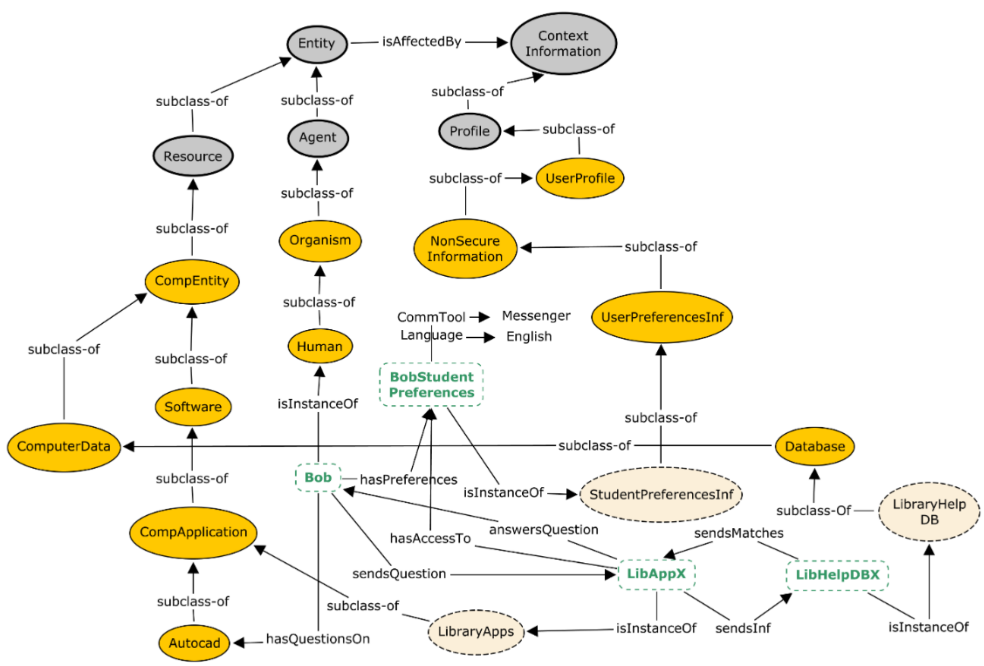


Fig. 23 Library service – scenario 6

The ontology representation of the above scenario (see Fig. 23) uses 16 classes of the upper and middle levels of the proposed model to represent domain classes such as StudentsPreferencesInf, LibraryHelpDB, etc., needed to represent different instances of the domain that collaborate among them to automate a library service. In this case, a library app can access to the preferences of different students and based on this information provide a better library service.

Scenario 7 - Health scenario. "*John is affected by a chronic disease. Her wife Barbara and his daughter Emily live with him and provide daily assistance services. John's home is equipped with a context-aware system, consisting of: monitoring devices (biomedical and environmental sensors), emergency and ordinary call buttons, and a PC which collects and analyses sensed data…*" [181][226] (see Fig. 24).

The ontology representation of the above scenario (see Fig. 24) uses 39 classes of the upper and middle levels of the proposed model to represent instances of a resource, agent, etc., and domain classes such as PatientHealthPlan and HospitalBuilding needed to represent the health plan of a human and a hospital respectively. In general, the instances are related to illustrate the interaction of a PC placed in a home with a server placed in a hospital to generate a health plan of a patient.

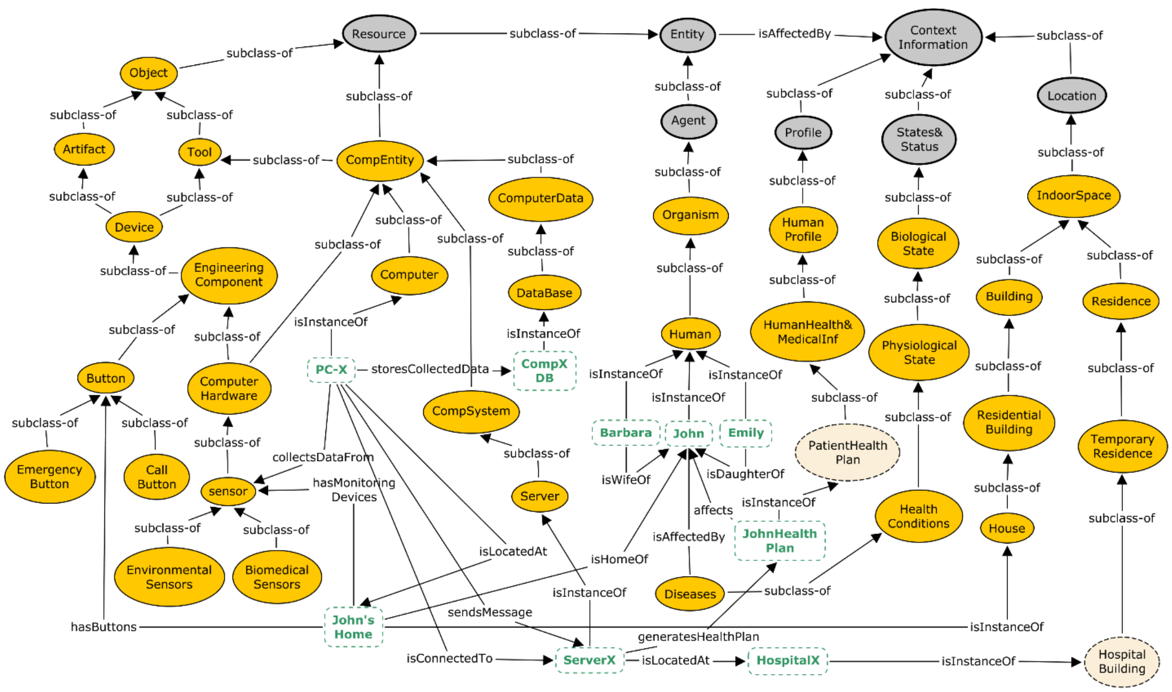


Fig. 24 Patient health and smart home – scenario 7

Scenario 8 - A scenario of smart space (smart meeting room). *"On Tuesday morning, a presentation about pervasive computing is scheduled to take place from 14:00-17:00 in the No.2 meeting room which is a smart meeting room in the No.1 building. The day before the presentation, the system has sent the meeting schedule including the title, the speaker, start time and location…"* [151] (see Fig. 25).

The ontology representation of the above scenario (see Fig. 25) uses 31 classes of the upper and middle levels of the proposed model and such classes are extended by domain classes needed to instantiate different context information and entities that play an important role in the scenario. In general, the instances are related to illustrate how a smart meeting room manage a meeting. As it can be seen in the figure, a presentation has an interval of time and a profile describing the important information of the meeting such as place, title, etc. that can be accessed by a schedule app that can conduct the management of the meeting.

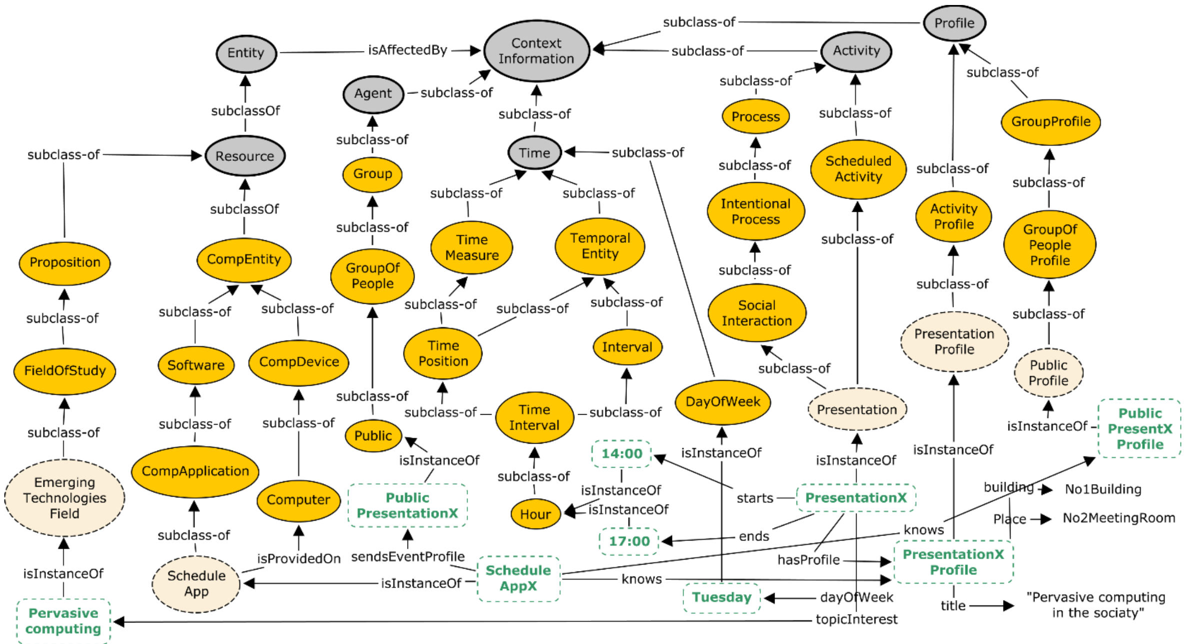


Fig. 25 Smart space ontology – scenario 8

Scenario 9 - Healthcare domain. "*The scenario begins with patient Bob who is in the emergency room due to a heart attack. While not being Bob's usual treating physician, Jane, a medical practitioner of the hospital, is required to treat Bob and needs to access Bob's emergency medical records from the emergency room…*" [104][227][228] (see Fig. 26).

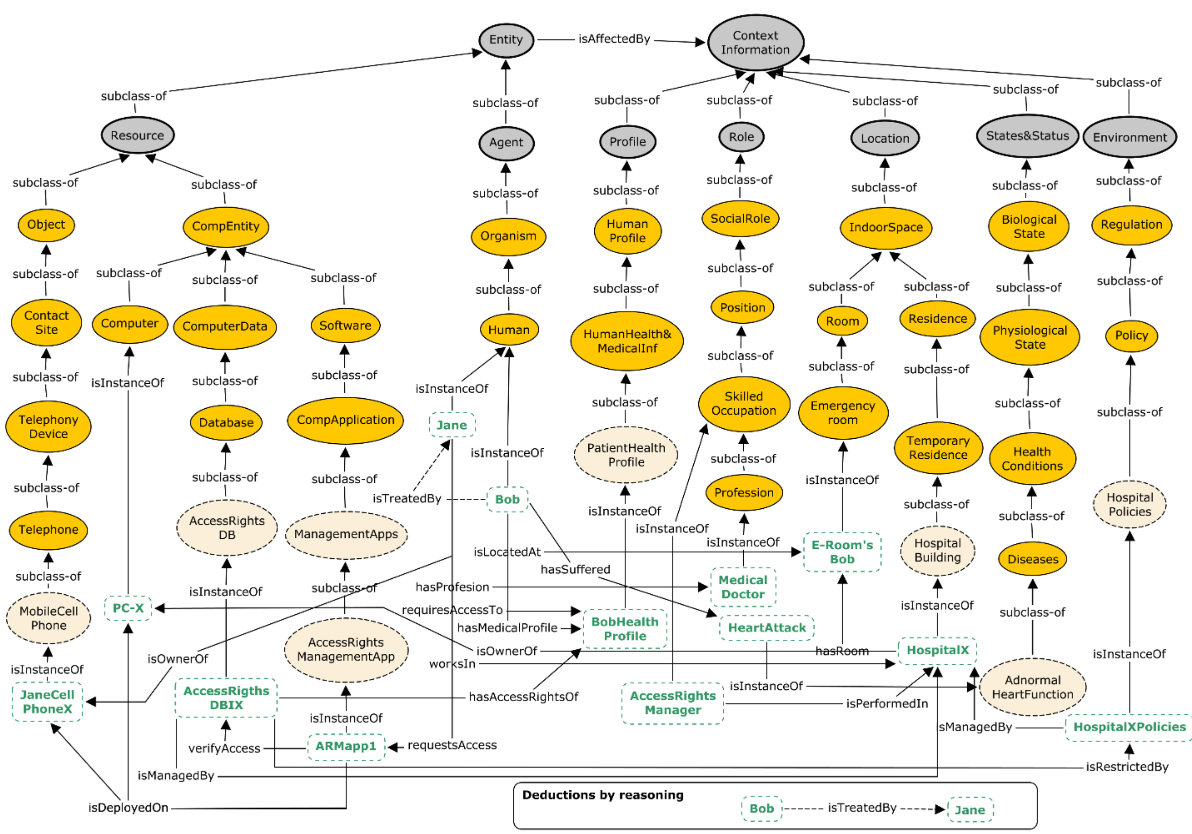


Fig. 26 Healthcare ontology – scenario 9

The ontology representation of the above scenario (see Fig. 26) uses 38 classes of the upper and middle levels of the proposed model, and 75 domain classes needed to instantiate different context information and entities that play an important role in the scenario. In general, the instances are related to illustrate the management of access rights to medical profiles such as who can provide access, how can be conducted the access request, where the access rights are deployed, what are the policies that restrict the access rights, etc.

As it can be seen in the figure, Bob is a human located in an emergency room because he has suffered a heart attack. Jane that is a doctor requires access to the Bob health profile in order to assist him. For this purpose, Jane makes an access request to the application called "ARMapp1" that manages the access rights of the personal working on the hospital. If this access that is restricted by the policies of the hospital can be assigned to Jane, then Jane can treat Bob. Note in Fig. 26 that such fact is asserted by the ontology through its reasoning capabilities. For instance, it can be asserted through semantic rules that can evaluate dynamic context (e.g., policies that can change dynamically given the changes of another context information, emergency in the state and status of the patient, etc.).

Scenario 10 – Smart call scenario. "*The user is sleeping in the bedroom or taking a shower in the bathroom, incoming calls are forwarded to voice mail box; when the user is cooking in the kitchen or watching TV in the living room, the volume of the ring is turned up…*" [19] (see Fig. 27).

The ontology representation of the above scenario (see Fig. 27) uses 30 classes of the upper and middle levels of the proposed model, and 5 domain classes needed to represent different domain instances that play an important role in the scenario. In general, the instances are related to illustrate the management of a call service. As it can be seen in the figure, a phone is controlled by a PC that collects data from a camera who detects the current state of a human that has a current state and location in his home.

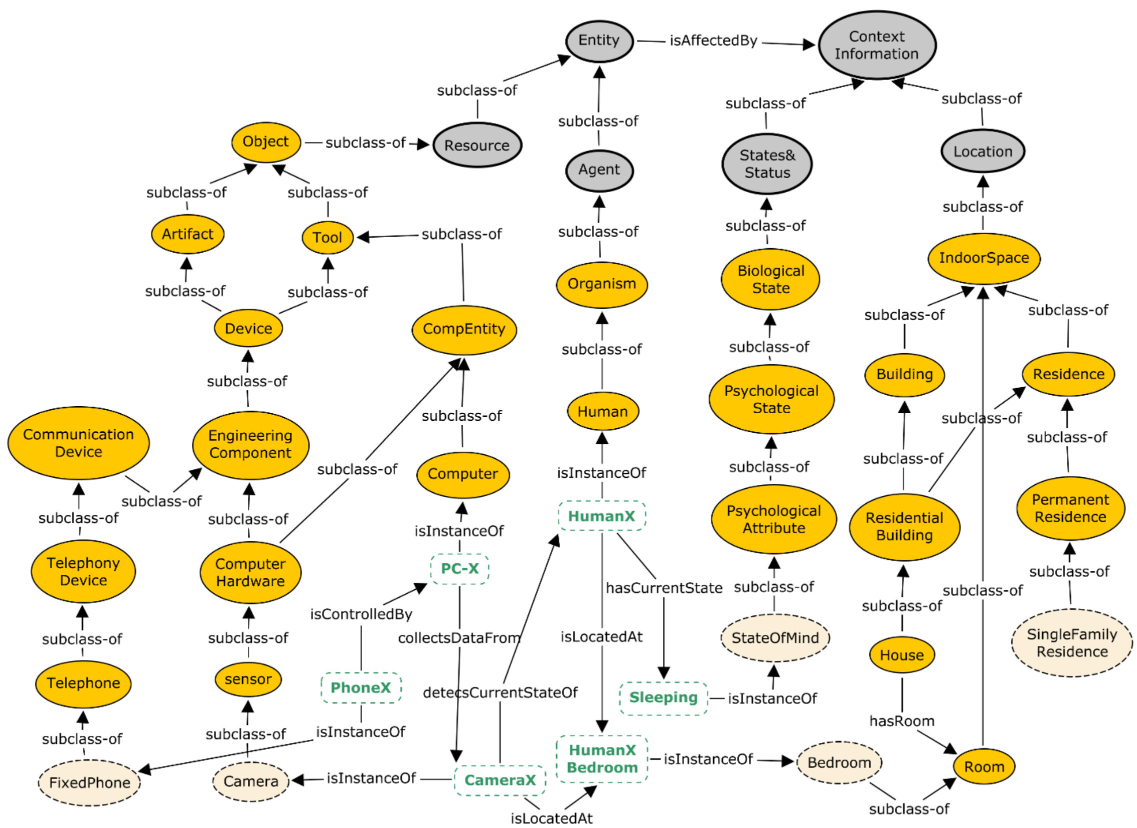


Fig. 27 Smart call ontology – scenario 10

Scenario 11 – Smart home scenario. "*Daddy John carrying a cell phone has entered his house; the face recognition system senses his presence and his location information get updated. When John moves into the bathroom to take a shower or goes to his bedroom for a nap…"* [229] (see Fig. 28).

Similarly to the previous scenario, the ontology representation of the above scenario (see Fig. 28) uses 29 classes of the upper and middle levels of the proposed model, and 5 domain classes needed to represent different domain instances that play an important role in the scenario. In general, the instances are related to illustrate the management of a smart home through faces recognition. As it can be seen in the figure, a PC has a system that controls different sensors and through them, the system can recognize the face of a human and based on this recognition the system can trigger different actions.

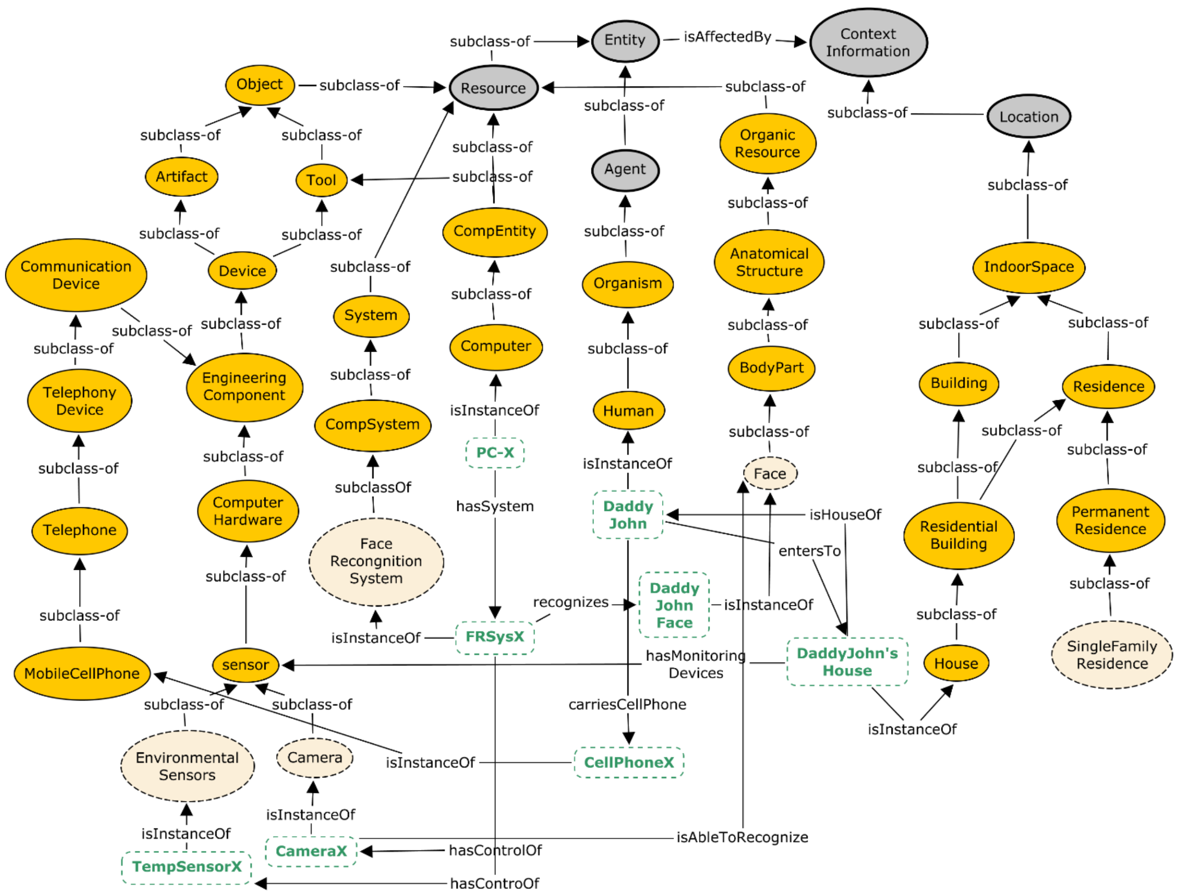


Fig. 28 Smart home ontology – scenario 11

Scenario 12 – Smart restaurant service. *"A restaurant could eventually provide different suggestions depending on who is walking by. If parents with children walk by, the restaurant would show the children's menu; if a couple is looking at it in the evening, it would show the menu for a candle light dinner; and if it is hot and sunny in the afternoon, the restaurant would advertise the selection of ice cream."* [23] (see Fig. 29).

The ontology representation of such scenario can be illustrated as depicted in Fig. 29. As it can be seen in the figure, a restaurant named Azurmendi is managed by a booking system that is installed in a PC-X. When a human makes a booking to the restaurant, the system requires some information provided by the human who made the reservation or it can be detected based on other source of information. As an example of usage, in the next section we made a query to this domain ontology to obtain direct or deduced context information.

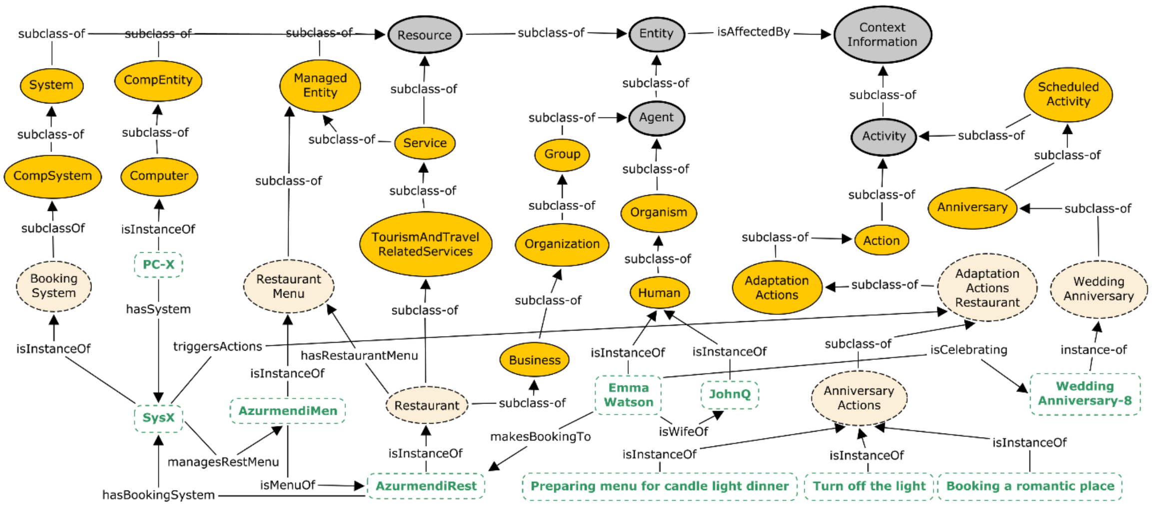


Fig. 29 Smart restaurant service ontology – scenario 12

Scenario 13 – Smart driving context. In this case, consider that an ontology modeler wants to characterize common primitives of a smart driving context. For this purpose, the modeler decides to reuse an existing domain ontology that can already provide the needed vocabulary. Hence, the modeler decided to reuse the OCM ontology [230] that in general has the aim to conceptualize traffic context and sensor capability. However, although the ontology provided the needed vocabulary at domain level, it was not aligned with a foundational or generic ontology in order to formulate and provide a consistent proposal that can increase the clarity of the terms used, the generality for improving the knowledge sharing, the uniformity for improving interoperability, etc. At this respect, the modeler decided to align the OCM ontology with the ontology proposed in this thesis for also increasing and unifying the semantic of OCM (see Fig. 30).

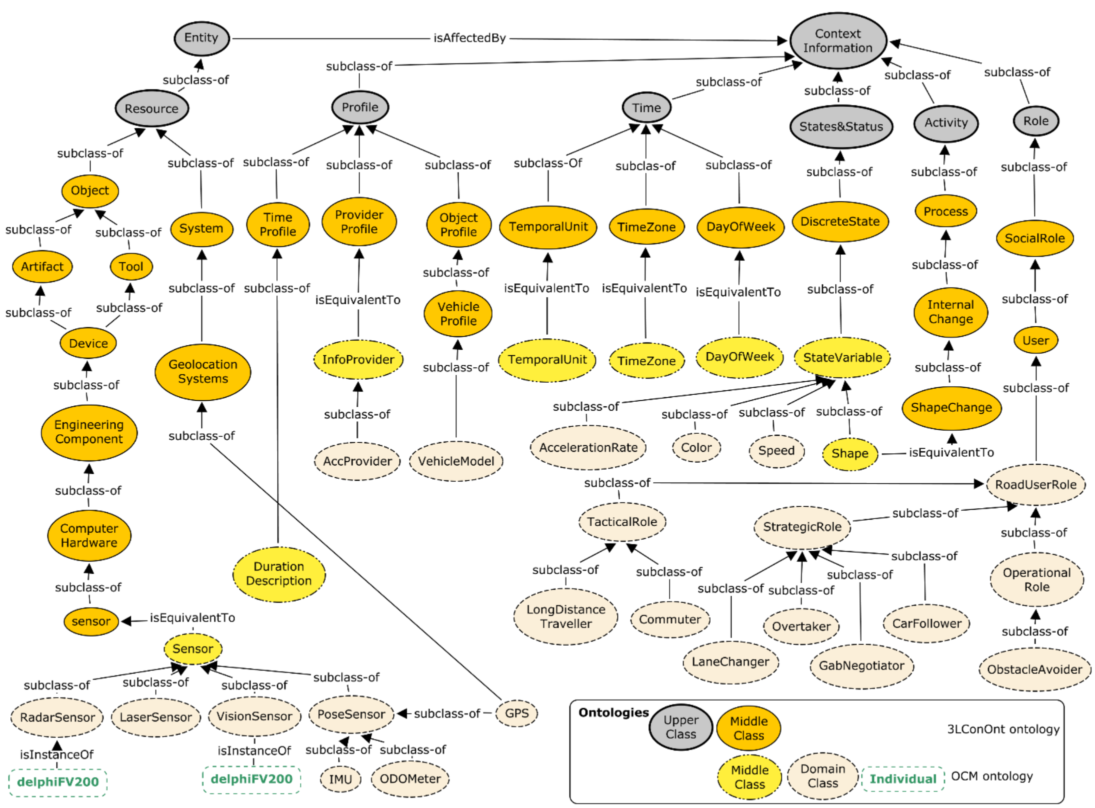


Fig. 30 Smart driving context – scenario 13

As it can be seen in Fig. 30., the type of scenario described above involves a bottom-up approach, i.e., an existing domain ontology is mapped to an existing abstract ontology. For this purpose, the modeler has the responsibility to 1) map the equivalent classes between both models to avoid inconsistencies, for instance, in the figure, 6 classes were mapped as equivalent classes of the proposed ontology; and 2) map the classes of the domain ontology as subclasses of the abstract ontology to increase the consistency of the final model, for instance, in the figure, 4 classes were mapped as subclasses of the proposed ontology. It is worth noting that although most of the modelers avoid this mapping process due to the time consuming derived mainly from the process of getting acquainted with an existing ontology, we consider that this cost of time can be balanced by increasing the interoperability and sharing knowledge of the final model among context-aware services.

Through these 13 scenarios, we have demonstrated the generality, extensibility and reusability of the proposed context model. Highlighting that although such scenarios of the existing contributions were represented in an ontology, they lacked of a structure representing a rich semantic of the entities and context information playing an important role in the scenarios. The scenarios also show that the proposed model, specifically the upper and middle level ontologies, can be adapted in several applications domains namely smart parking, smart home, smart health, smart restaurant service, etc. To do so, the model represents three levels of abstraction, where the upper and middle levels of the ontology specify context knowledge useful for representing and structuring domain ontologies in a rich and formal semantic increasing their reusability and applicability in different projects of the context-aware computing.

3.4.2 Evaluating usage, consistency and functionality

In this section, we validate the usage, consistency and functionality of the proposed model by triggering reasoning and queries over the model and modules. Specifically, to conduct such reasoning and queries tasks, we use some modules of the middle-level ontology such as Location, States&Status, Activity, etc. We also implement two domain ontologies to carry out these tasks such as the presented in Section 3.3.3 and the smart restaurant service illustrated in Fig. 29 because it manages different entities and context information that can be consulted in different use cases of a smart restaurant. The proposed model is implemented in Protégé and delivered as follows:

- The upper-level ontology and the modules of the middle-level ontology are implemented separately for allowing selective reuse of the ontology parts that apply to every particular scenario.
- The domain ontology depicted in Fig. 17 is implemented for illustrating the role of the lower-level ontology and supporting the validation addressed in this section. Hence, at this level of the model different domain ontologies can be implemented and linked to the middle and upper levels of the model.
- Each implemented level or module provides its specific context knowledge pieces, i.e., its classes, individuals, object and datatype properties that directly affect such level or module of the ontology.
- All implemented levels and modules are integrated allowing powerful context reasoning.

The OWL of each level and module of the proposed three-level ontology for context modelling can be found in https://github.com/ocabgit/Three-LevelContextOntology.git.

The reasoning capabilities of a context ontology are defined as the ability of deducing new knowledge, and understanding better, based on the available context [231]. According to Wang et al. [19], the use of context reasoning has two goals: checking the consistency of context; and deducing high-level, implicit context from low-level, explicit context. Based on the second feature, we can infer the location of a person, the environment of a location, the state and status of a person or device, etc. We demonstrate this potential reasoning process on the proposed ontology considering two types of reasoning: *ontology-based reasoning* and *rule-based reasoning*. The *ontology-based reasoning* uses the existing reasoning rules already defined in the semantics of OWL (e.g., rdfs:subClassOf, owl:sameAs, etc.). To illustrate *ontology-based reasoning*, the transitivity rule of the isAffectedBy property infers the following situational context of a Human: if Chris is affected by the temperature of certain location and this temperature is also affected by the pollution of the environment, then Chris is also affected by the pollution of the environment (see Table 23).

Table 23. Ontology reasoning about the environment of an entity

Explicit context	*Implicit context*
<owl:ObjectProperty rdf:ID="isAffectedBy"> <rdf:type="owlTransitiveProperty"/> </owl:ObjectProperty> <Human rdf:ID="Chris"> <isAffectedBy rdf:resource="#Temperature"/> </Human> <EnvConditions rdf:ID=" Temperature"> <isAffectedBy rdf:resource="#Pollution"/> </EnvConditions >	<Human rdf:ID=" Chris"> <isAffectedBy rdf:resource="#Pollution"/> </Human >

On the contrary, *rule-based reasoning* is not included in the semantics of OWL and therefore, the rules should be explicitly defined by users. One of the main use cases of these rules is finding the match among them and the context information retrieved (e.g., from sensors) to perform an action or deduce complex context. In this regard, consider the following rules defined in Protégé:

1. *Human(?h), Automobile(?a), hasLocation(?h, ?a), performsActivity(?a, Accelerating), hasStatusState(?a, On) -> performsActivity(?h, **Driving**)*
2. *Office(?o), MobileCellPhone(?m), Human(?h), Activity(?ac), Agenda(?a), hasLocation(?h, ?o), hasStatusState(?m, Status-Off), hasScheduledActivity(?a, ?ac) -> hasStatusState(?h, **Working**)*

These rules infer the activities of a person, if all the statements are true, i.e., if all the context information retrieved match with the rule then it is deduced an activity. For instance, in the first rule it is specified that if a "Human" is located in his "Automobile" and the "Automobile" has the status 'On' but also it is "Accelerating", then a "Human" is "Driving". Similarly, the second rule specifies that if a "Human" is located in his "Office", has his "Agenda" with an activity scheduled and his "Mobile" has status 'Off' then the "Human" is "Working". Such rules can be specified in two ways, by using the Semantic Web Rule Language (SWRL)[14] into the Protégé editor or by means of Jena, a Java framework for developing Semantic web applications incorporating a rule reasoner API.

In this part of the thesis, we formulate rules in Protégé and Jena for validating the capabilities of the context model. The *rule-based reasoning* is also supported by Jena since it provides a Java Rule object with a list of terms (premises), a list of head terms (conclusions), and an optional name or direction. From this perspective, the syntax of Jena for defining rules is different from the syntax used in Protégé as previously specified. For illustrating this fact, we translate the first rule defined above by employing the syntax of Jena as follows:

1. *[(?h rdf:type Human), (?a rdf:type Automobile), (?h :hasLocation ?a), (?a :performsActivity Accelerating), (?a :hasStatusState On) -> (?h :performsActivity **Driving**)]*

For triggering queries to the model, we make available the 3LConOnt in the following URI: http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt*/module#resource*. As it can be seen, such URI is composed by */module#resource*, where *module* is the name of the module that is needed

[14] http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

in the query (e.g. ComputationalEntity), and *resource* is the class, instance, object and datatype property that is needed in the query. For instance, a specific URI for a query would be as follows: http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/ComputationalEntity#Software.

We also implemented a query engine by using the Jena API along with the SPARQL[15] query language. This query engine validates and assists the interaction with the three-level ontology by querying the context information that characterises the situation of entities such as services, user, provider, etc. Furthermore, the reasoning capabilities of the query engine are given by the Apache Jena framework, thus the context information gathered is also deduced by considering transitive and inferred relations. One of the methods for this purpose is the OntModelSpec.OWL_MEM_MICRO _RULE_INF containing a transitive reasoner which can be used to infer properties such as subClassOf and subPropertyOf. Therefore, the query engine is able to gather direct and indirect descendants of context information. In Fig. 31, a fragment of the console results when querying the subclasses of the Location module from the context model is shown. As it can be seen, if the query engine does not use reasoning capabilities, the subclasses of Location are only the direct ones, otherwise both direct and indirect subclasses are considered.

```
Without reasoning
---------------------------------------------------
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#IndoorSpace>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#LocationCoordinates>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#OutdoorSpace>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Region>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#RelativeLocation>
---------------------------------------------------

Using reasoning capabilities
---------------------------------------------------
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#IndoorSpace>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#LocationCoordinates>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#OutdoorSpace>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Region>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#RelativeLocation>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Building>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Cave>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Garage>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Laboratory>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Office>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Location#Automobile>
---------------------------------------------------
```

Fig. 31 Reasoning capabilities of the query engine component

In addition, the context model proposed in this thesis is tested below by considering the use case scenario of the smart restaurant service described in Section 3.4.1 (scenario 12). Such scenario can be addressed as follows: the query engine takes as input the name of a client "Emma Watson" that is required to be characterized by means of context information. Based on this input, the query engine requires the information from the context ontology that returns simple and deduced context information. Given the evaluation and analysis of the entities and its corresponding context information performed by the context ontology, the actions for adaptation purposes of the restaurant service are presented in Fig. 32.

[15] http://www.w3.org/TR/sparql11-query/

Entities and Context information

<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Resource#AzurmendiRest>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Resource#AzurmendiMen>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Agent#EmmaWatson>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Agent#JohnQ>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/Activity#WeddingAnniversary-8>

Actions for adaptation purposes

<Preparing menu for candle light dinner>
<Booking a romantic place>
<Turn off the light>

Fig. 32 Use case validation – smart restaurant service

Finally, we have executed some queries to the proposed model using the SPARQL into the Protégé editor. Such queries are related to the competence questions specified in Section 3.3.3 and the expected results should validate that layers and modules of the model, the extension and integration tasks, and the conceptualization of the domain ontology are correct and consistent. The queries are depicted as follows.

In Fig. 33 is depicted a query to obtain all the monitoring tools that supervise the Twitter social network. As it can be seen, two types of monitoring tools are retrieved namely SocialMentionAPI and TwitterAPI. In Fig. 34 a variation of the query specified in Fig. 32 is depicted to obtain only the monitors that supervise Facebook. As it can be seen, only the SocialMentionAPI has such capability.

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?twittermonitors
WHERE {?twittermonitors <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#isMonitoringToolOf>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/ComputationalEntity#Twitter>}
```

twittermonitors
SocialMentionAPI
TwitterAPI

Fig. 33 SPARQL query of Twitter monitoring tools

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?facebookmonitors
WHERE {?facebookmonitors
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#isMonitoringToolOf>
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/ComputationalEntity#Facebook>}
```

facebookmonitors
SocialMentionAPI

Fig. 34 SPARQL query of Facebook monitoring tools

Fig. 35 shows the monitored data that can be obtained from a social network monitoring tool. As it can be seen, the timestamp is the generic monitored data that is related to a list of data items namely dataitemID, dataItemsLink, dataItemMessage, etc.

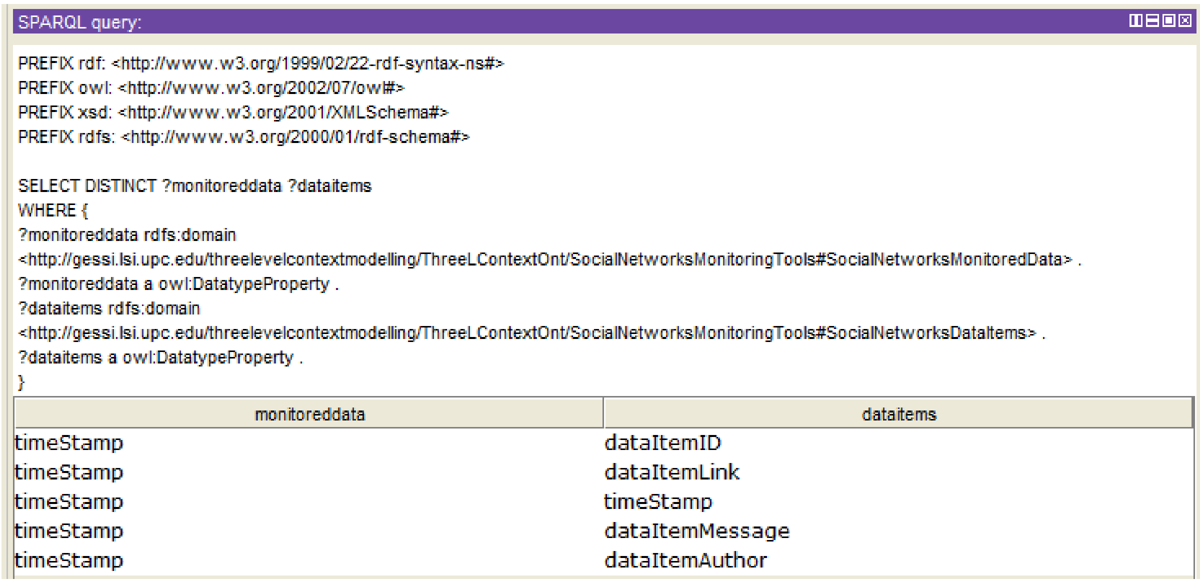


Fig. 35 SPARQL query of monitored data and data items related

3.4.3 Evaluating reusability

As previously mentioned, the levels and modules of the proposed model can be reused independently of the entire model. In other words, an ontology designer can integrate in an existing ontology the level of abstraction or module that is required without the dependency of other levels or modules of the model, since the semantic of each level and module is independent of each other. Such capability is validated here by illustrating into the Protégé tool the reuse of a level of the model in an existing ontology.

Consider the following case. The SOUPA ontology provides different ontologies that can be reused in different contextual domains. However, they lack of a semantic structure that can be given by an upper-level ontology. For solving this issue, we suggest to reuse the upper-level ontology proposed in this thesis. For instance, we opened in Protégé the Person ontology provided by SOUPA (see Fig. 36).

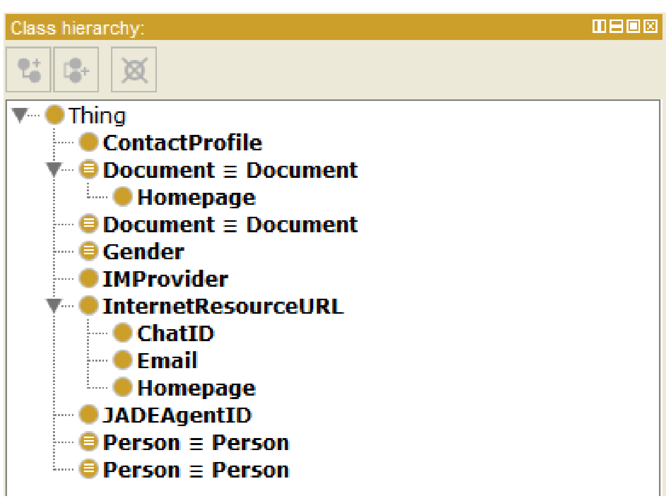


Fig. 36 Person ontology of SOUPA

As it can be seen in the figure, the Person ontology of SOUPA provide a vocabulary and a class hierarchy that can be complemented or restructured with a rich semantic. For this reason, we imported the proposed upper-level ontology into the Person ontology of SOUPA (see Fig. 37). Hence, the class hierarchy depicted in Fig. 36 was restructured as depicted in Fig. 38 to provide better semantic of the proposed vocabulary in the SOUPA ontology. In this regard, we can say that the contact profile is a context information that belongs to a generic class named profile that can describe an agent through a resource.

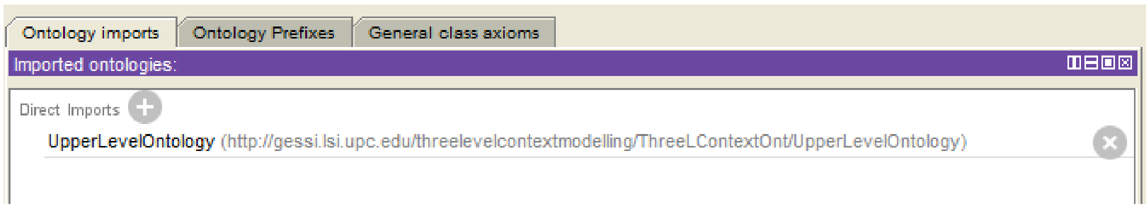


Fig. 37 Importing the upper-level ontology

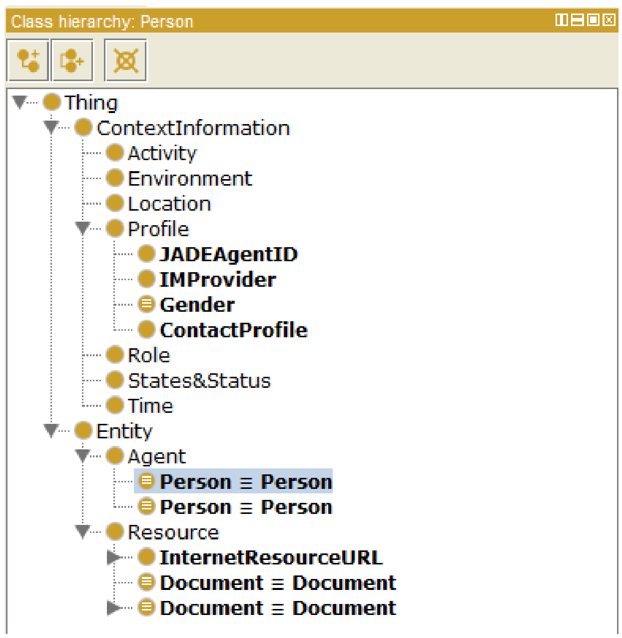


Fig. 38 Rich semantic of the person ontology by reusing the upper-level ontology

3.4.4 Evaluating correctness and completeness

The details and results of this validation are specified as follows:

A) Correctness. To carry out this validation we rely on the advises given in [232] that specifies that *"a basic requirement for a modular ontology to be correct is that each module is correct"*. This is the first perspective that we have adopted to conduct the correctness validation of the proposed ontology. Second, we have conducted a syntactic correctness, consistency among the specified primitives, and consistency between instances and specifications through automatic tools as it is also recommended. For this purpose, we have used 1) the Pellet reasoning engine available in Protégé (see Fig. 39). Note that some rules were used in this evaluation to improve the validation; 2) a validation service[16] provided by the W3C where the ""triple" view (RDF) of the proposed model was also evaluated (see Fig. 40); and 3) RaDON [233] plugging of NeOn Toolkit[17] that has the aim of verifying inconsistencies and incoherencies in ontologies (see Fig. 41). As it can be seen,

[16] https://www.w3.org/RDF/Validator/

[17] http://neon-toolkit.org/wiki/Main_Page.html

the results obtained in this validation process showed that none inconsistencies were found in any of the modules and layers of the proposed ontology. Note the dotted squares in each figure. In Fig. 39 “reasoner active” means that none inconsistencies were found by the reasoning engine.

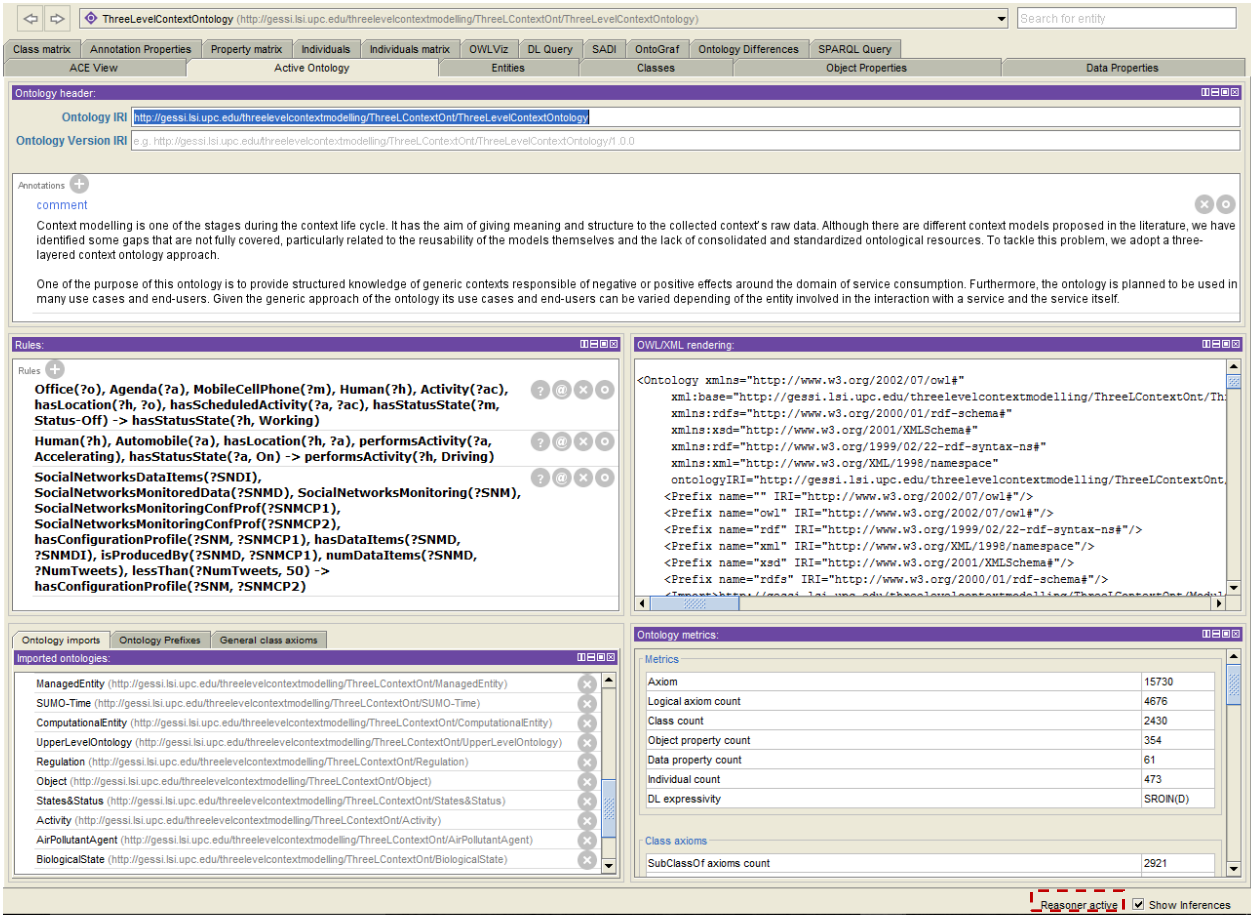


Fig. 39 Validating correctness through the reasoning engines of Protégé

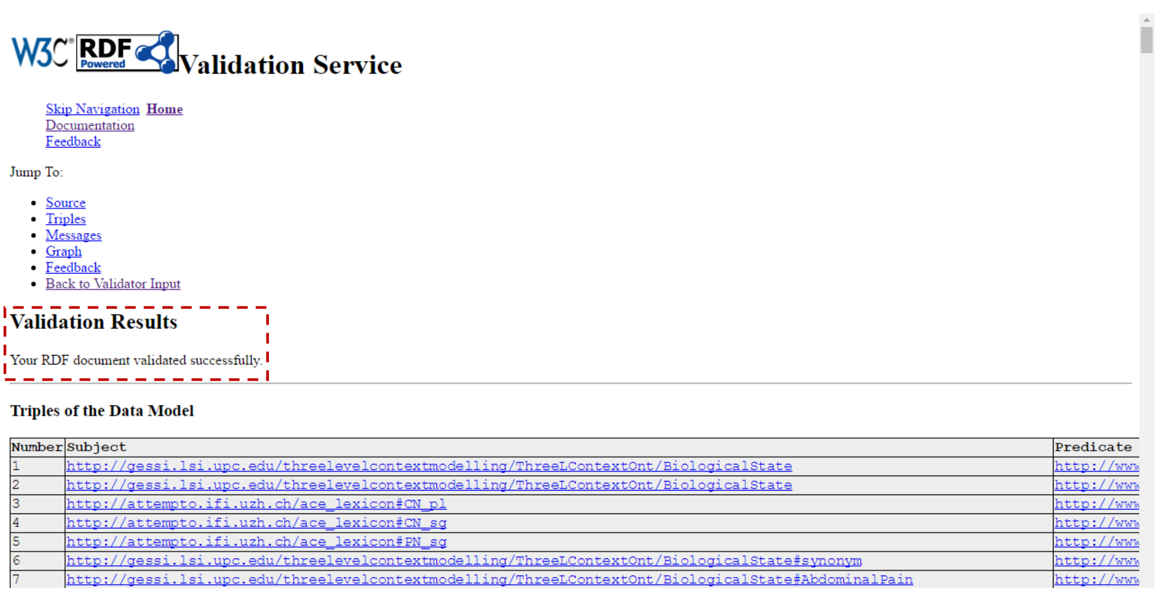


Fig. 40 Validating correctness through the validation service of W3C

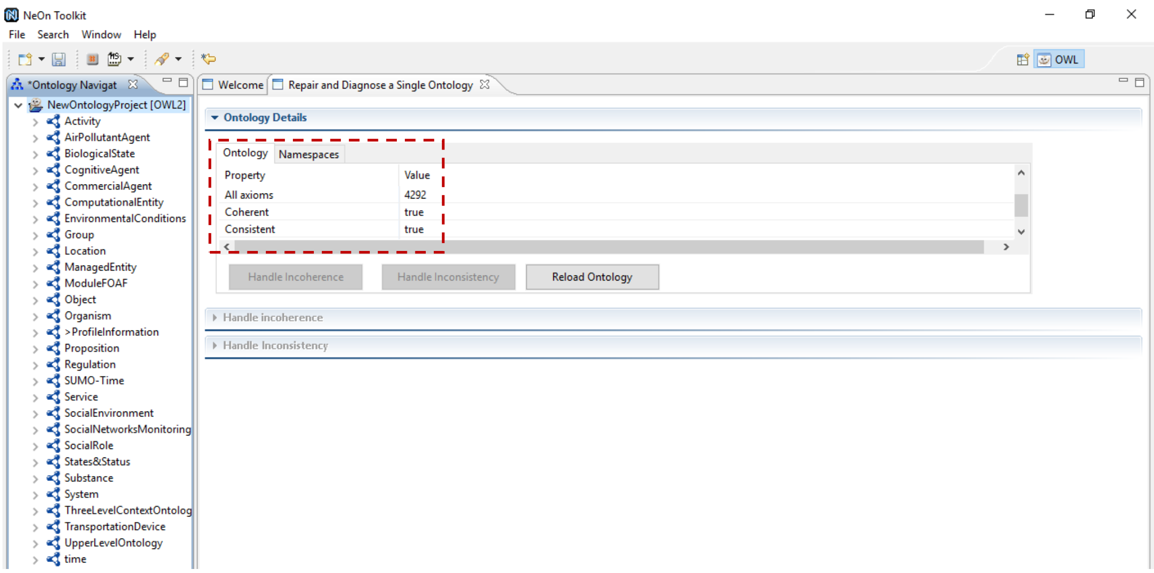


Fig. 41 Validating correctness through RaDON plugging of NeOn Toolkit

B) Completeness. At this respect, it is important to highlight that although the proposal of this thesis takes as a basis 8 representative ontologies that were commonly referenced in the context modelling research area, the context knowledge, semantics and patterns of the proposal is stated also by considering the 138 contributions that were selected in the systematic mapping study addressed in Chapter 2. From this perspective, we consider that the completeness of the context knowledge represented in the proposal of the thesis has been improved with respect to the knowledge represented in the analyzed contributions. Hence, to carry out this validation we have performed a comparative analysis of completeness between the proposal and the 138 proposals of context modelling (see Table 24). As it can be seen, we follow the same criteria represented in Table 12 of Chapter 2 adding the following criteria of percentages: 100% of completeness for ✓*, 90% of completeness for ✓, 80% of completeness for P*, 70% of completeness for P, and 0% of completeness for X. As we have expected, the results of the completeness study show that our proposal has achieved 100% of completeness in the knowledge represented in the main modules of the upper-level ontology, i.e., we explicitly provided an structured vocabulary for Agent, Resource, Activity, Time, Environment, Location, Role, and States and Status.

Table 24. Validation of completeness

Proposals	**Agent**	**Resource**	**Activity**	**Time**	**Environment**	**Location**	**Profile**	**Role**	**States & Status**	***Completeness***
Henricksen et al. [66]	P	P*	X	X	X	X	X	X	X	18.89
Held et al. [67]	P	P*	X	X	✓	X	✓	X	X	38.89
Tarasewich [68]	P	P	✓*	✓*	✓*	✓	P*	X	✓*	81.11
Ghannem et al. [69]	P	P*	X	X	✓	P*	X	X	X	40.00
Mok and Min [70]	X	X	✓	✓	X	✓	P	X	X	37.78
Chen et al. [17]	✓	P*	P*	✓	P	✓	✓	X	X	70.00
Cao et al. [12]	P	P	✓	✓	P	✓	P*	P	X	72.22
Devaraju and Hoh [71]	✓*	✓*	✓	X	P	✓	P*	P	P	76.67
Pietschmann et al. [72]	✓*	P*	✓*	✓*	X	✓*	✓	✓	✓	85.56
Cadenas et al. [73]	P	P*	X	✓	✓	✓	P*	✓	P	77.78
Rubio et al. [74]	P	X	P	✓	✓*	✓*	P*	P	X	66.67
Cabrera et al. [26]	X	✓	✓	✓	✓	✓	✓	✓	X	70.00
Gu et al. [6]	✓	P*	✓*	X	P	✓*	P*	X	X	62.22
Hofer et al. [78]	P	P*	X	✓	X	✓	P*	X	X	50.00

Proposals	Agent	Resource	Activity	Time	Environment	Location	Profile	Role	States & Status	*Completeness*
Preuveneers et al. [79]	P	✓*	✓	✓	✓*	✓*	✓*	✓	P*	93.33
Heckmann [80]	P	P	X	X	P	P	P*	✓	P*	63.33
Chen et al. [81]	✓*	P*	✓*	X	X	✓*	P*	✓*	X	66.67
Xynogalas et al. [82]	P	P*	X	✓	P	✓*	✓	X	✓*	68.89
Bu et al. [83]	✓*	P*	X	X	X	X	P*	X	X	33.33
Kim and Choi [84]	✓*	P*	✓	✓	P	✓*	P*	X	X	72.22
Neto and Pimentel [85]	✓*	P*	✓*	✓*	P	✓*	✓*	✓	X	84.44
Park and Kwon [86]	P*	P*	✓	✓	✓	✓	P	✓	X	80.00
Ngo et al. [88]	✓*	P*	✓*	✓*	✓*	✓*	✓*	X	X	77.78
Bradley and Dunlop [89]	P*	P*	P*	P	✓	P	P	P	P	82.22
Kritsotakis et al. [90]	P*	P*	P	X	X	✓*	✓	P*	P*	73.33
Sheng et al. [91]	P*	P*	✓*	X	P	✓*	✓	X	X	62.22
Lee and Kwon [93]	X	P	X	X	P*	X	X	X	P	26.67
Chaker et al. [94]	P	X	P	X	✓	X	P*	P	P	52.22
Hervás et al. [96]	P*	P*	P*	X	P*	✓*	✓	✓	✓	85.56
Najar et al. [47]	✓*	✓*	P*	✓	P	✓	P*	✓	X	82.22
Prekop et al. [97]	✓	✓*	✓*	X	X	X	P	X	P	47.78
Khedr and Karmouch [98]	✓*	P*	✓	✓	P	✓*	✓	✓	P	88.89
Gong et al. [99]	P*	✓*	P	X	X	X	P*	P*	X	52.22
Villegas and Müller [100]	✓*	P*	✓*	✓*	P	✓*	P*	X	P	82.22
Strassner et al. [101]	✓*	✓	✓*	✓*	P	✓*	P*	P	X	81.11
Zimmermann et al. [102]	✓*	P*	✓	✓	P	✓	P*	P	P*	90.00
Shen and Cheng [103]	✓	✓*	✓*	✓	✓	✓*	P*	P*	X	85.56
Kayes et al. [104]	✓*	✓	P	✓	P	✓	✓	✓	✓	86.67
Śliwa and Gleba [105]	P	P*	P*	X	X	X	✓	P	X	47.78
Krömker and Wienken [106]	P	X	X	X	P*	✓	P	P	X	44.44
Aly et al. [107]	P	P*	X	X	X	✓*	X	X	X	30.00
Khemaja and Buendía [108]	P*	P	✓*	✓	P*	✓*	P	P*	X	81.11
Hoque et al. [109]	P*	P*	✓*	✓*	P*	✓*	X	P*	X	77.78
Hameurlaine et al. [110]	X	P*	X	✓	P*	✓	✓*	P*	X	64.44
Khouja and Juiz [111]	P	✓	X	X	X	X	X	P*	X	28.89
Liang-Liang [112]	X	✓	X	X	X	X	P*	X	X	21.11
Saidani et al. [113]	P	✓*	P	✓	✓*	✓	P*	✓	X	67.78
Restrepo et al. [114]	X	P	P	P	P*	P	P	P	P	65.56
Lee et al. [115]	X	X	P*	✓	P*	P*	X	X	P	51.11
Xue et al. [116]	X	P*	X	X	P*	P*	P	X	✓	51.11
Wang and Tang [117]	✓	✓*	X	X	X	X	P*	X	P	40.00
Sohn et al. [118]	P	X	X	✓	✓*	P	✓	X	X	46.67
Yus et al. [119]	P	X	✓*	X	X	✓*	X	X	X	30.00
Choi et al. [120]	X	P*	X	P*	✓*	✓*	X	X	P	41.11
Sotsenko et al. [121]	X	P*	✓	✓	✓*	P*	P*	X	P	72.22
Li and Chen [122]	X	X	X	X	X	P	P*	✓	X	28.89
Chellouche et al. [123]	P	P*	P	P	✓	X	✓*	P	X	63.33
Zhang et al. [124]	X	P*	P*	✓*	X	✓*	P*	X	P	63.33
Berri [125]	P	P*	✓*	✓	X	✓*	P*	✓	X	72.22
Li et al. [126]	P	P*	✓	✓	✓*	✓	X	X	X	60.00
Katsumata [127]	P	X	✓	P	X	X	P	X	X	33.33
Da et al. [128]	X	P	X	X	X	P	P*	X	X	26.67
Benlamri and Zhang [129]	P	P*	✓*	X	✓*	✓	P*	P	X	70.00
Miraoui [130]	P	✓*	X	X	X	X	P*	P	X	37.78
Rahman et al. [131]	X	X	✓*	✓*	X	✓*	P*	✓*	X	55.56
Diallo et al. [132]	X	✓	✓	X	✓	✓	P*	P	P	66.67
Maalej et al. [133]	P	P*	✓	✓	✓	✓	✓	P*	X	80.00
Furno and Zimeo [134]	✓*	X	X	✓	P	✓*	P	✓	X	57.78

Proposals	Agent	Resource	Activity	Time	Environment	Location	Profile	Role	States & Status	*Completeness*
Novakovic and Huemer [135]	P	X	✓*	X	P	P	P*	X	X	45.56
Ai et al. [136]	P*	P	X	✓*	P*	✓*	P*	X	P	60.00
Boudaa et al. [137]	✓	P*	✓	✓	✓*	✓	✓	X	X	72.22
Kim et al. [138]	P*	P*	✓*	✓*	X	X	P*	X	P	63.33
Fissaa et al. [139]	P	P*	X	✓	P	✓	P*	P	P*	76.67
Zouhaier et al. [140]	P	P	✓	X	✓	X	P*	P	X	54.44
Kabir et al. [141]	P	X	P	X	X	X	P*	P	✓	44.44
Paul.and Wachsmuth [142]	P	X	✓	X	P	✓	P	X	X	43.33
Hu et al. [143]	P	X	✓	✓	P	✓	P*	P	X	64.44
Copeland and Crespi [144]	X	X	✓	✓	P	P	P*	X	X	46.67
Wong et al. [145]	✓	P*	✓*	X	X	✓*	P*	P	X	62.22
Jianfeng and Dong [146]	P*	P*	X	✓*	P*	✓	P	X	X	62.22
Madkour and Maach [147]	P	P*	✓	X	P*	✓	✓	P	P*	78.89
Jang and Choi [148]	P	✓*	✓*	✓*	P	✓*	✓*	X	X	71.11
Gilman et al. [149]	P	P*	P	✓	P*	✓	✓	P	P*	86.67
Wusheng et al. [150]	P*	P	✓	X	P	✓	P*	P	P	73.33
Li [151]	P*	P*	✓*	✓	P*	✓*	P*	P	X	84.44
Kim et al. [152]	P*	P	P	X	✓	P	P*	P	X	63.33
Changboka et al. [153]	X	✓	✓	✓*	X	X	✓*	P	X	50.00
Durán et al. [154]	P	X	X	X	✓*	✓	✓*	P	✓	57.78
Tong et al. [155]	P	P*	X	X	P	✓	P*	X	P	55.56
Luo et al. [156]	P	P	X	✓	P	X	P*	X	X	44.44
Yi et al. [157]	P	P*	P	X	✓	X	P*	P	P	63.33
Sun et al. [158]	P	P*	✓	X	✓*	X	P*	P	P	66.67
Sigg and David [159]	P*	P*	✓*	✓*	✓*	✓*	✓	✓	P*	97.78
Park et al. [160]	P	P*	✓	X	P	✓	P*	X	P	65.56
Zainol and Nakata [161]	P*	P*	✓	X	P*	✓	✓	X	P	71.11
Saeedi et al.[162]	P	✓	✓*	X	X	✓*	P*	X	X	51.11
Bandara et al. [163]	X	P*	X	X	X	X	P*	X	P	30.00
Liu et al. [164]	P	P*	P	✓	✓*	✓*	P	X	✓*	77.78
Hu and Li [165]	P	P*	X	✓	✓*	✓	P*	P*	P*	83.33
Cioara et al. [166]	✓	✓*	X	X	P	P	P*	P	X	55.56
Cao et al. [167]	P*	P*	✓	X	P	✓	P*	X	P	68.89
Lamsfus et al. [168]	X	P*	✓	✓	P	✓	P*	P	X	67.78
Hur et al. [169]	P	P	✓	X	P	✓	✓	X	X	53.33
Coma et al. [170]	X	X	P	✓*	P	P	P*	P	X	53.33
Tan et al. [171]	P*	P*	✓*	✓	✓*	✓*	✓	✓	✓	95.56
Sun et al. [172]	P	P*	P	✓	✓*	✓*	P*	P	P*	88.89
Ouyang et al. [173]	✓*	P*	P	X	✓*	X	P*	P	X	60.00
Ko and Sim [174]	X	P*	✓*	X	X	✓*	✓	X	X	43.33
Chang et al. [175]	X	X	X	X	X	X	X	X	✓*	11.11
Ghadiri et al. [176]	P	P*	X	✓	P	✓	P*	X	P*	68.89
Erfianto et al. [177]	P*	P*	✓	X	P	✓	P*	X	X	61.11
Qin et al. [178]	P*	P*	✓*	✓	✓	✓	P	P	X	78.89
Lee and Meier [179]	X	X	X	✓	P	✓	P	X	P	43.33
Mei et al. [180]	X	✓	X	X	P	X	P	P	X	33.33
Paganelli et al. [181]	✓*	P*	✓	X	✓	✓	P*	P	X	71.11
Hwang et al. [182]	P*	P*	P*	✓*	P*	✓	✓	P*	P	94.44
Choi and Yoon [183]	P*	P*	✓	✓	✓	✓	✓	P	X	80.00
Hwang et al. [184]	P	X	P	X	P*	✓	✓	P	X	54.44
Jrad et al. [185]	X	P	X	X	X	✓	P*	X	P	36.67
Pessoa et al. [186]	P	P*	✓*	X	X	✓	X	X	X	40.00
Hamdeh and Ma [187]	P	P*	✓*	✓	X	✓*	X	X	P*	62.22
Strimpakou et al. [188]	✓	P*	✓*	✓	X	✓	P*	X	P*	74.44

Proposals	Agent	Resource	Activity	Time	Environment	Location	Profile	Role	States & Status	*Completeness*
Sheng et al. [189]	X	P	P	X	P*	P	P*	P	P	61.11
Wang et al. [190]	P	P*	P	X	P	P	P	P	P	65.56
Yang et al. [191]	P*	P*	P	✓	✓*	✓*	✓*	P	P*	92.22
Kranenburg et al. [192]	P	P*	P	X	✓	P	P	P	P	67.78
Chaari et al. [193]	P	✓	✓	X	P	✓	P*	X	X	56.67
Ou et al. [194]	✓*	P*	✓*	✓	P	✓*	P*	P	X	81.11
Almeida et al. [195]	P	X	P	P	P	✓	✓*	P	P*	71.11
Hu and Moore [196]	P	P	✓	✓	P	P	P	✓	✓*	80.00
Amundsen and Eliass. [197]	X	P*	X	X	X	X	P*	X	P*	33.33
Hong et al. [198]	P*	P*	✓	✓	✓*	✓*	P	X	X	72.22
Kalyan et al. [199]	✓	P*	P	✓	X	✓*	P*	P	P	76.67
Go and Sohn [200]	P*	P*	X	X	✓	P	P	P	P	63.33
Zacarias et al. [201]	P	X	P	✓	X	X	X	✓	X	35.56
Ko et al. [202]	P	P*	✓	X	✓*	X	✓	X	P*	61.11
Kaenampor. and Neill [203]	P*	P*	P	✓	P	✓	✓	✓	X	77.78
Korpipaa et al. [4]	P	P*	✓	✓	✓*	✓	P*	X	X	71.11
Our proposal	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	100

"A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it"

- Frank Herbert

CHAPTER 4

How to acquire: building a monitoring architecture

Context acquisition plays an important role in the context life cycle since it is responsible for (1) gathering the context raw data from different entities (agents and resources) and (2) disseminating the gathered data to the interested parties and consumers. Context acquisition is usually implemented through monitoring tools that integrate software services and sensors as data gathering instruments. There are several monitoring tools proposed in academia and industry, which mainly focus on providing information of the quality of service (QoS) of the monitored entities.

This contribution of the PhD thesis is focused on how to acquire context from the perspective of a **context-aware monitoring architecture**, which basically opens two main challenges to solve related to the two responsibilities of context acquisition (data gathering and dissemination). First, context-aware monitoring tools should support dynamic capabilities, i.e., reconfiguration and adaptation to continuously obtain and provide reliable context information. This characteristic is necessary to bind the behavior of the monitoring tool to the behavior of the context-aware system under supervision. If the system incorporates new features or changes the behavior of existing ones, the monitoring tool needs to be able to gather different data. In addition, a context-aware monitor needs to react seamlessly when a source of data becomes temporarily unavailable. Second, the gathered data needs to be classified according to a well-defined data schema in order to avoid risks in data management, mainly in its dissemination and storage, due to ambiguity, inconsistencies and poor integrity and characterization.

To tackle these research challenges, we present in this contribution of the PhD thesis a Context-Aware Monitoring Architecture (CAMA). It is based on state-of-the-art software patterns of service-oriented architecture, enabling to plug and play different monitors that can be easily reconfigured, evolved or adapted for supporting highly dynamic environments. CAMA includes a component to handle the context ontology already introduced in Chapter 3, specifically the domain context ontology depicted in Fig. 17. Such ontology can support the conceptualization of inputs, outputs and capabilities of monitoring tools that can be useful to disseminate a unified and representative schema of monitored data. CAMA also includes a component to handle different types of monitoring tools. At this respect, such component is supported by a state of the art of monitoring tools that has the aim of providing an extended view of features and approaches of different tools that can be plugged in CAMA. Additionally, since each of these tools exhibits its own characteristics that can make it more adequate than others in a concrete situation. We also propose a first version of a quality model for analyzing monitoring tools in the domain of Web services since this type of services are becoming the cornerstone of modern service-based software applications [234].

4.1 Research questions of RQ3

To address systematically and contribute solving the research challenges of this contribution of the thesis, we have divided the main research question RQ3 specified in Table 1 as follows:

- RQ3.1: How can we provide a context-aware monitoring tool with the required reconfiguration and adaptation capabilities to face the constantly changing situation of the services and applications that compose a context-aware system?
- RQ3.2: How can we structure, store and manage the data gathered by a context-aware monitoring tool in order to be easily disseminated to other components?

The RQ3.1 can be also decomposed into the following sub-RQs, considering the typology of events that may trigger a reconfiguration.

- RQ3.1.1 How can a context-aware monitoring tool manage the supervision of new features or entities?
- RQ3.1.2 How can a context-aware monitoring tool manage dynamic capabilities (i.e., reconfiguration, adaptation and/or evolution) when new requirements or changes in the context data, functions or quality features of an entity are detected?
- RQ3.1.3 How can a context-aware monitoring tool manage the failure of a data gathering instrument?

4.2 State of the art of context-aware monitoring infrastructures

In the last recent years, several research approaches have emerged in the field of context-aware monitoring. Most of the approaches are aligned with trending computer paradigms, e.g. mobile and ubiquitous computing, Smart Cities and the Internet of Things. We have evaluated the related work by analyzing how they address the RQs previously defined. For each RQ, the different proposals are evaluated with the marks ✓, ~ ,✗ following the condition criteria defined in Table 25.

Table 25. Evaluation criteria for the analysis of the state of the art

RQ3.1.1	
✓	Adding new monitors can be done with minimal effort
~	Adding new monitors can be done, but requires some effort
✗	The system does not provide the capability to add new monitors, or the monitors that can be added are limited to a specific scope
RQ3.1.2	
✓	Monitors can be reconfigured automatically
~	Monitors can be reconfigured with some effort or the reconfiguration of monitors is limited
✗	The system does not provide the capability to reconfigure monitors
RQ3.1.3	
✓	The failing monitor can be replaced automatically
~	The failing monitor can be replaced with some effort, or the detection of failures is limited
✗	The failing monitor cannot be replaced
RQ3.2	
✓	Data can be accessed using a rich semantics method following an open-world assumption
~	Data can be accessed using a rich semantics method following a closed world assumption, or the architecture allows methods of gathering data following an open-world assumption, but this aspect is not fully covered or investigated
✗	Data is provided without rich semantics through an API or similar

The selection of papers has been conducted through a systematic search described as follows: we have searched for papers containing the keywords “context” and “monitoring” in the last 10 years, from 2006 to 2016, using the Google Scholar database. Due to the enormous amount of results, we picked the 25 most relevant papers according to the database ranking. From these 25 results, 18 were discarded for being out scope, leading to 7 papers to analyze. Then, we conducted a snowballing process to obtain relevant papers not included in these first 25 results. In this step, we included the works that were cited by more than one paper and satisfied the same criteria of being from 2006 to 2016 and in the scope of context monitoring. Only 2 new papers were added, yielding to the final results of 9 papers to be analyzed as shown in Table 26 and described in depth below.

Table 26. Analysis of related work

Proposal	RQ1			RQ2
	RQ1.1	RQ1.2	RQ1.3	
MobiCon [235]	✓	✓	~	~
CARE [236]	✓	✓	✗	~
Orchestrator [237]	✓	✓	~	✗
SeeMon [238]	✓	✓	✗	~
ContextProvider [239]	✓	✓	~	✗
CoMon [240]	✗	✗	✓	✗
HiCon [241]	✓	~	~	~
CLAD [242]	✗	✗	✓	✗
Contory [243]	✓	✗	✓	~
CAMA (our proposal)	✓	✓	✓	✓

RQ3.1.1. Most of the approaches have an extensible architecture capable of adding new monitors to gather new metrics [235][236][237][238][239][241][243]. The ones that do not provide this capability is because their scope is limited to a specific set of sensors in a particular environment [240][242].

RQ3.1.2. Most approaches can automatically reconfigure the monitors dynamically [235][236][237][238][239]. Nevertheless, some approaches include only static sensors without parameters that can be dynamically reconfigured [241], or such reconfiguration requires some effort as they were not intended to change at runtime [240][242][243].

RQ3.1.3. Most approaches have a mechanism to replace failing monitors by means of an extensible architecture as well as mechanisms to support dynamic addition of monitors. However, few of them include automatic means to detect if a monitor is failing. In some cases, the system can detect when a monitor is unavailable but other types of failures are not investigated [237]. On the contrary, Contory [243] can detect when a sensor is unavailable or fails, to replace it automatically. Similarly, CoMon [240] and CLAD [242] have multiple sensors and then, if a sensor fails, data is replaced by data from nearby sensors.

RQ3.2. Most approaches provide just an API to gather the monitored data without rich semantics [237][239][240][242]. Those that provide a richer query mechanism to gather the data, follow a close-world assumption (i.e. they assume that all data is present in the system) [235][238][241][243], which is not the most suitable approach in context-aware monitoring. Only CARE [236] follows an open-world assumption. However, although CARE’s architecture could eventually interact with ontological reasoners, this aspect was not addressed in the paper.
To conclude, as depicted in Table 26, none of the existing approaches satisfies all three aspects of RQ3.1, and only CARE [236] has the potential of satisfying RQ3.2. Even more, no single approach satisfies more than 2 out of the 4 aspects tackled by RQ3.1.1, RQ3.1.2, RQ3.1.3 and RQ3.2.

4.3 A software architecture for context-aware monitoring tools

We present in this section CAMA (Context-Aware Monitoring Architecture), a software architecture able to integrate different individual monitoring tools. CAMA is intended to support the constant changes in the context that characterizes the situation of different entities including not only the services and applications that conform the context-aware system under supervision, but also its own components.

To design CAMA, we have taken into account the following high-level requirements, mainly derived from the RQs presented in Section 4.1 of this chapter.

- **Req1:** CAMA must provide the capability to add new monitors with minimal effort. This requirement addresses RQ3.1.1.
- **Req2:** CAMA must provide the capability to reconfigure the monitors automatically. This requirement addresses RQ3.1.2.
- **Req3:** CAMA must provide the capability to replace a failing monitor automatically. This requirement addresses RQ3.1.3.
- **Req4:** CAMA must provide the capability to access the monitored data using a rich semantics method following an open-world assumption. This requirement addresses RQ3.2.
- **Req5:** CAMA must facilitate its integration with other systems.
- **Req6:** CAMA must allow the integration of several monitoring tools that monitor different software components or devices.
- **Req7:** CAMA must be secure in terms of authentication and authorization.

Note that the last three requirements do not come directly from any RQ, but we consider them necessary in order to make the resulting architecture usable in practice. To satisfy these requirements, CAMA is structured in a loosely coupled architecture that is organized as follows (see Fig. 42):

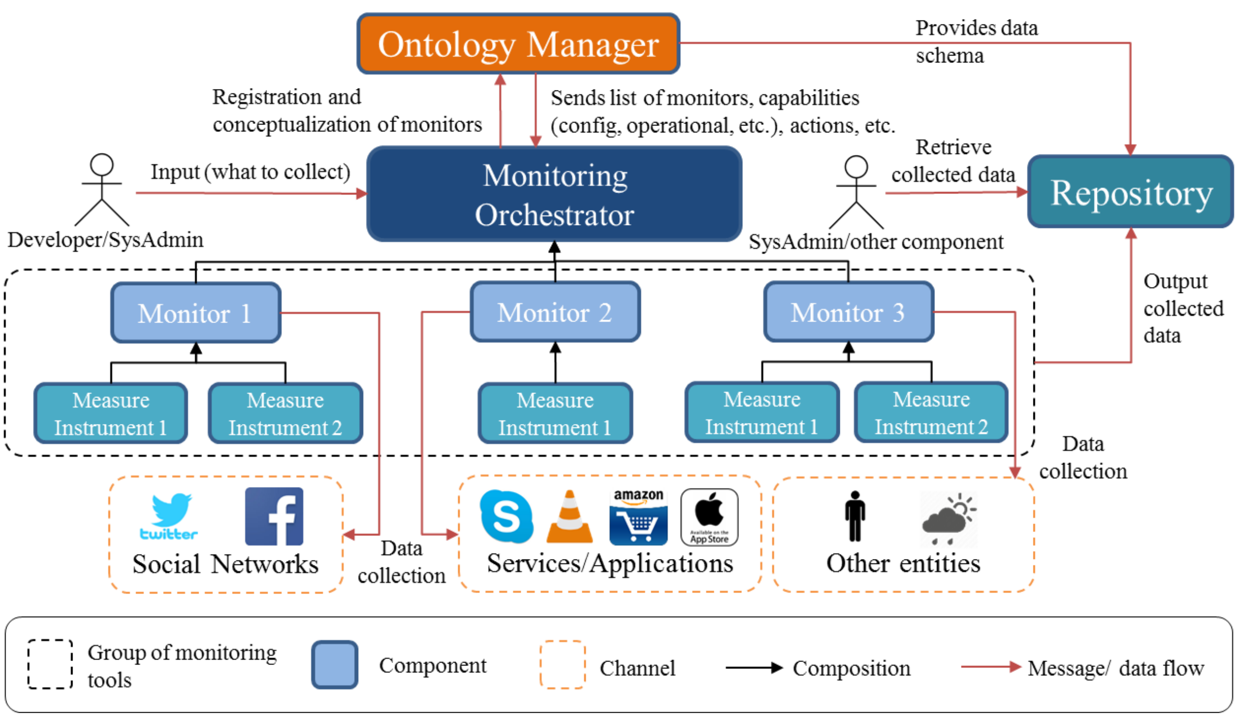


Fig. 42 Context-aware monitoring architecture (CAMA)

The resulting architecture is a service-oriented architecture (SOA). We consider that adopting this paradigm and applying the typical SOA principles and patterns [244] is the way to ensure *Req5*. Just to illustrate the consequences of such decision in CAMA, the different APIs of the monitoring tools should be wrapped as micro-services for providing highly decoupled services performing small tasks. Such perspective allows the reconfiguration of the monitors, satisfying *Req2*. Finally, to avoid unauthorized access, fraudulent components, etc. the communication between components can be done through an Enterprise Service Bus (ESB), which can easily contain a security module that handles authentication and authorization, satisfying *Req7*. The components of CAMA satisfying the remainder requirements are described in the next subsections.

4.3.1 Group of monitors

This set of architectural components represents the monitoring tools that gather context-related data, supporting then *Req6*. Each monitor component in the architecture represents a monitoring tool, which is composed by one or more measure instruments that implement a specific logic to gather data from software services, applications and physical devices. The measure instruments perspective of the architecture is useful when monitoring tools provide different APIs to retrieve different aspects of software systems (e.g. Nagios monitoring tool [245]). A summary of the state of the art that provides an extended view of monitoring tools is provided in subsection 4.3.1.1 and a quality model for analyzing monitoring tools is provided in subsection 4.3.1.2.

4.3.1.1 State of the art of monitoring tools and approaches

We supported the approach of group of monitors by evaluating the current state of the art of academic and commercial monitoring contributions. Specifically, we intended to answer the following research questions: 1) which are their tracking techniques; 2) which are their configuration capabilities; 3) which are their extension and integration capabilities; 4) which entities or artifacts are monitored; 5) which contextual data is monitored; 6) how violations are identified and managed; and 7) how monitored data is exploited or disseminated.

Table 27 summarizes the results obtained from the evaluation of the selected monitoring tools (50 academic and 33 commercial). An extended report of the specific results of each research question, and the main features and references of each monitoring contribution is provided in [30].

Table 27. Summary of results of the state of the art of monitoring tools and approaches

Research questions	Summarized results
1) Which are the tracking techniques used in monitoring contributions?	• The tracking techniques identified were mostly related to passive and active monitoring, and intrusive and nonintrusive monitoring • A small set of monitoring approaches can apply multiple monitoring techniques • Most of the identified approaches support passive monitoring • Only a few approaches address event-driven data collection and other isolated techniques • Security issues were found in most of the monitoring techniques
2) Which are the configuration/reconfiguration capabilities provided?	• Most of the approaches do not perform some kind of configuration • Another set of the approaches although have specified a kind of configuration do not clarify if it is performed statically or dynamically • Most of the approaches that perform configurations are conducted at runtime • A small set of the approaches conduct static configurations

3) Which are the extensibility and integrability capabilities provided?	• Most of the academic monitoring tools provide integration capabilities • Most of the open source monitoring tools provide integration capabilities • For reusing purposes the monitor is mainly provided as a component or plugin
4) Which kind of services, applications or artefacts can be monitored?	• Academic monitoring approaches are mainly focused on supervising Web services • Commercial monitoring approaches are more complete solutions that supervise different artefacts • A small set of monitoring approaches are focused on supervising social networks • An important set of monitoring approaches are focused on supervising distributed or local applications as well as network and video/audio streams
5) Which contextual data, metrics, information or aspects from QoS or QoE, can be monitored?	• Most of the information monitored is related to QoS • Most of the QoE information monitored is based on QoS metrics • Only a small set of monitoring approaches supervise context information and implicit user feedback • Most of the commercial monitoring tools supervise multiple information
6) How is the identification of violations performed and how can this be managed?	• Most of the monitoring approaches providing a proactive monitoring technique also consider a reactive technique • Most of the violations are identified through the service level agreements (SLAs) • Most of the proactive techniques relies on periodical testing
7) How is the monitored information delivered, reported and made available?	• Most of the commercial monitoring tools provide reports, dashboards, statistics, etc. but remote access to the information monitored is not allowed • Some approaches incorporate an API to share the monitored information • Only a small set of monitoring approaches stores in log files the monitored information

4.3.1.2 A quality model for analyzing monitoring tools

The state of the art showed a variety of monitoring tools with different approaches and capabilities that can be used in different application context. Such diversity of monitoring tools suggests the need of an analysis framework to select the most appropriate monitoring tool for a particular need that can be useful from the perspective of group of monitors defined in CAMA. For this purpose, we have developed a quality model by applying the Individual Quality Model Construction (IQMC) [246] that is defined upon the ISO/IEC 9126-1 quality standard [247]. As a result, we provided essential information, which may help in the following issues: monitoring tools selection in a particular situation; knowledge about the features that monitoring tools offer; quality attributes that are fundamental for monitoring tools; etc.

The proposed quality model is depicted in Table 28. It is a first version that is focused on just one quality characteristic, which in fact is one of the most important characteristics since it describes the functionality of the monitoring tools.

Table 28. Quality model for analyzing monitoring tools

Characteristic	Subcharacteristic	Basic attribute	Attribute level 1
Functionality	Suitability	Quality attributes	Built-in
			Attribute Management
		Monitor	Monitor Management
			Monitoring Strategies
			Monitoring Notifications
			Monitoring Reports
		Users	User Management
			User Actions
		Web services	WS Management
			WS Monitoring
		Agreements	SLA Management
			SLO Management
	Accuracy	Verifiableness	History Control
			Data Versioning
			Logging Capabilities
		Effectiveness	Self -Test Results
			Published Test Results
	Interoperability	Direct	By Means of Protocols
			By Means of APIs
		Indirect	Indirect Interoperability
		Monitor	Monitor Architecture
			Monitor Platform
	Security	Application	Provided by the Application
			Provided by Third Parties
		Data	Data Stored
			Data Transmitted
		Monitor	SOAP Request Security
			SOAP Response Security

The attributes identified in the model (see Table 28) are described as follows:

- *Quality attributes suitability*. Built-in registers the metrics offered by the monitoring tool. Attribute management records if there is the chance to add new attributes to be monitored by the tool.
- *Monitor suitability*. Monitor management synthesizes the different actions that may be undertook with the monitor, e.g. creation of Widgets (to have easy access), configuration of alarms, etc. Monitoring strategies specify the active or passive monitoring strategies, as mentioned in Table 27. Monitoring notifications to configure what happens when violations are detected. Monitoring reports to declare the informational capabilities of the tool (statistics, etc.).
- *Users suitability*. Includes two attributes for recording the supported user management operations (user management) and the user actions that are allowed (user actions).
- *Web service suitability*. Defines two attributes WS management and WS monitoring for defining how WS may be made available to the monitor and which information from them is necessary or advisable for proper monitoring.
- *Agreements suitability*. States two attributes SLA and SLO management for defining how SLA and SLO are dealt by the monitor.
- *Verifiableness*. History control states if the tool is able to provide a history of the changes on the data managed. Data versioning defines the ability to keep track of versions of the same data. Logging capabilities declares the ability or not to keep track of issues with a log resource (at least, a log file).

- *Effectiveness*. Self-test Results indicates if the tool offers some facility for testing its behavior. Published test results informs about the existence of test suites executed by the provider itself.
- *Direct interoperability*. Subcharacteristic that informs about the capability of the monitoring tool to directly interact with specified systems. It can be done by means of Protocols or by means of APIs (thus, two attributes).
- *Indirect interoperability*. Provides an attribute for describing the capability of a monitoring tool to interact with other systems by means of indirect mechanisms.
- *Monitor interoperability*. Refers to the monitor architecture and monitor platform of the monitoring tool to assess their degree of interaction with other applications.
- *Application security*. The main distinction is whether this security is achieved by using capabilities from the monitor itself (provided by the application) or by another software (provided by third parties).
- *Data security*. A distinction is made about the data: data stored and data transmitted.
- *Monitor security*. It is related to security of SOAP messages. SOAP request security states if the tool has some security mechanism that applies at the moment of transmitting a SOAP message. Its counterpart is SOAP Response Security.

4.3.2 Monitoring orchestrator

This component is responsible for first registering, and then orchestrating, the group of monitors integrated in the architecture. For this integration task, the monitoring orchestrator makes available an interface from which a developer or system administrator (hereafter, sysAdmin) can register and integrate monitors into the architecture, satisfying *Req1*. Then, the orchestration starts with the input of the developer/sysAdmin indicating “what to collect”, and based on this information the component will execute actions over the monitors and mechanisms that have been integrated into the architecture, such as applying assignments, changes, new configurations, etc., at runtime, allowing the monitors to be reconfigured automatically (see Group of monitors). Once the monitors are configured, they run independently of the orchestrator. In such a manner, we prevent to have a single point of failure in the system. If the orchestrator fails, only the (re)configurations would be unavailable. Nevertheless, for critical systems, this could be mitigated by replicating and deploying more than one orchestrator.

4.3.3 Ontology manager

This component provides and manages an ontology that characterizes the context of entities (agents or resources) through context information (time, location, environment, etc.). Thus, each monitor than can be considered as resource is registered and conceptualized in the ontology by considering the context information that the monitor is able to supervise. The registration and conceptualization tasks start when the orchestrator receives a subscription request of a new monitor with the following parameters: name, type, description, endpoint, inputs and outputs. The ontology manager maps such parameters into classes and properties (datatype and object properties) or creates an instance in case of such type of monitoring tool has already been modelled.

From this perspective, the ontology will know the list of monitors that have been registered, the context information that a monitor or mechanism is able to retrieve and provide (e.g., in the scope of the social environment, the messages of a certain social network), the operational and configuration capabilities of the monitors, the status of the monitors (by means of self-monitoring), as well as the actions that can be done over them. These actions that can be triggered by the reasoning capabilities of the ontology and by external components that are responsible for data analysis and decision-making have the aim of maintaining the health of both the entire architecture and the entities that are being supervised. Such mechanisms enable the detection and replacement of failing monitors, and hence, satisfying *Req3*. The ontology also provides the data schema of monitored data for a unified and structured dissemination, satisfying *Req4*. To allow the dynamic addition of monitors and provide the capability to deal with different monitored elements of heterogeneous monitoring tools, the ontology has been designed to be easily extensible and adaptable for each domain. More details about the ontology are given in Chapter 3, specifically in Section 3.3.3.

4.3.4 Repository

The repository component is responsible for storing the output of each monitoring tool that is structured by means of the data schema provided by the ontology through the ontology manager. The monitors know the structure of such schema and can communicate with the repository through the interface provided by this component, which in turn can be used by a SysAdmin or other external component to collect the monitored data, also satisfying *Req4*. The interface of the repository component also provides the methods required for instantiating it using different types of components. For instance, it can be instantiated with Kafka for messaging systems to capture and publish data, or MySQL for data persistence, among others.

4.3.5 Operational process and technical details of CAMA

The sequence diagram to configure and run the monitors is depicted in Fig. 43. The operational process of CAMA starts when a customer (a developer, the SysAdmin, etc.) makes requests to the monitoring orchestrator about what context information should be collected. At this stage, the *Orchestrator* should be aware of the monitoring tools that has available. Such task is supported by the ontology provided and managed by the *Ontology Manager*, which is responsible for registering and modelling inputs, outputs and configuration capabilities of the monitoring tools.

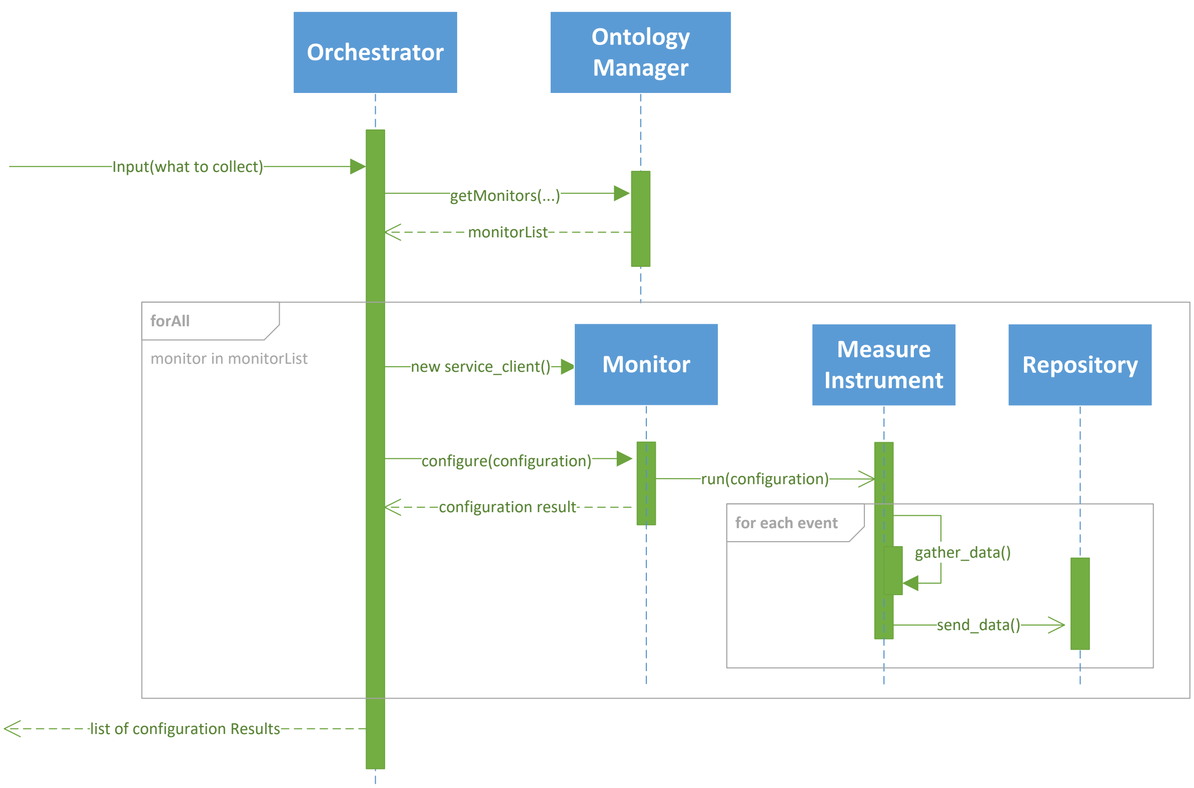


Fig. 43 Sequence diagram to configure and run the monitors

Hence, with the information provided by the ontology, the *Orchestrator* is able to perform different operations over the monitors such as turning on/off monitors, etc. Therefore, if a monitoring tool fails and there is another tool able to supervise the same context information (reasoning given by the ontology), the monitor that is not working is changed by another one. Once the available monitors are identified, the *Ontology Manager* will return them to the Orchestrator. The Orchestrator will then configure each *Monitor* accordingly, which in turn will run the *Measure Instruments* that periodically will gather and send the data to the *Repository*.

Concerning technological details, the *Monitors* to be plugged into the *Orchestrator* should be wrapped as a web service [248]. Finally, all the context information retrieved by the *Monitors* is stored in the repositories defined in the *Repository* component (e.g. MySQL, SQL, etc.). For disseminating and storing a unified and structured set of monitored data this component is also supported by the schema of the ontology that conceptualize the outputs of the monitoring tools.

4.4 Proof of concept: two instantiations of CAMA

In this section, we present a preliminary proof of concept for CAMA. We use the monitoring requirements of the SUPERSEDE European project (www.supersede.eu) and we consider two monitoring scenarios: social networks and IT infrastructure. Hence, specific monitors that retrieve the information required by the use case are instantiated in the architecture.

4.4.1 Monitoring social networks

It concerns the monitoring of comments on social networks for improving end-users' quality of experience of applications deployed in marketplaces and streaming services such as live webcasting of sport events. The monitoring architecture for the social networks use case is depicted in Fig. 44. We include three monitoring tools: Marketplaces, Twitter API [249] and Social Mention API [250]. Some of these monitoring tools implement more than one measure instrument: on the one hand, Twitter provides two APIs: the Streaming API and the REST API; on the other hand, we consider two different marketplaces with their respective APIs: Apple Store and Google Play.

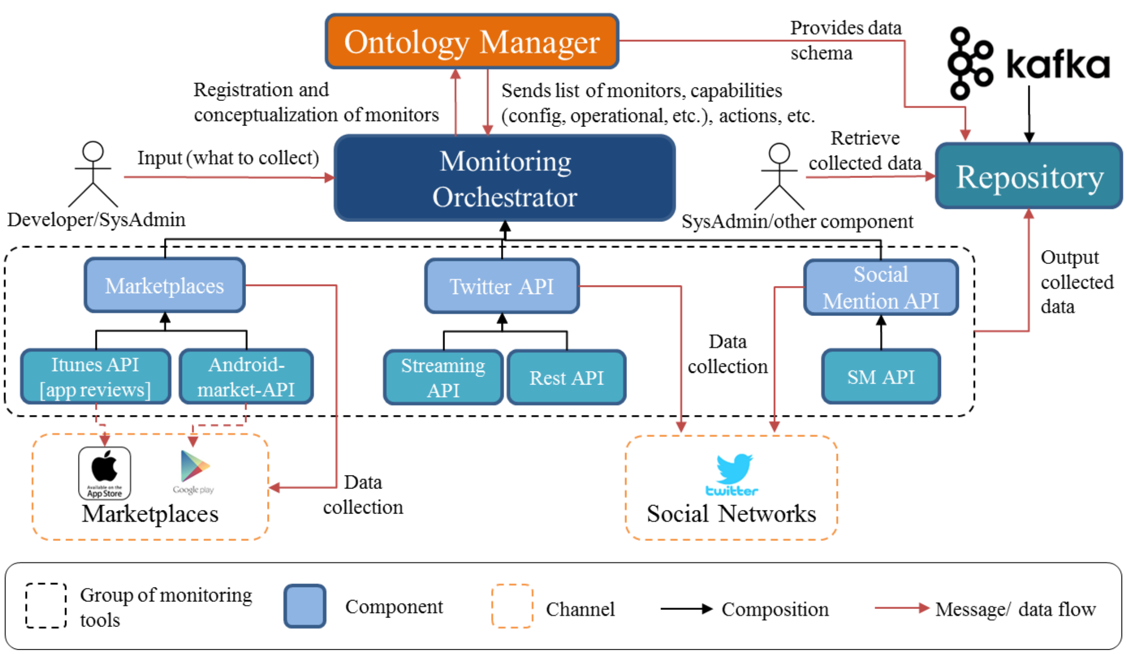


Fig. 44 Monitoring architecture: the social networks use case

The ontology handled by the ontology manager is the base of information of the whole architecture. It is aware of the inputs, outputs and configuration capabilities of the mentioned monitoring tools. With this information, the monitoring orchestrator can perform actions over the monitors. For instance, if the Twitter's streaming API fails, based on monitoring logs of each monitoring tool and the reasoning capabilities of the ontology, the orchestrator can change it by the REST API or even by the Social Mention API, which can feed the same required data in the monitoring infrastructure. Finally, the collected data by the instantiated monitoring tools is unified, structured and stored in Kafka, which is a publish-subscribe messaging system [251]. Hence, a SysAdmin or another external component can retrieve the collected monitored data from Kafka by means of the interface provided by the repository component.

4.4.2 Monitoring IT infrastructure

This case study concerns energy management and evolution of services and applications for improving end-users' quality of experience through monitoring the IT infrastructure including hardware, software, network resources and web events. The monitoring architecture for the IT infrastructure use case is depicted in Fig. 45.

As it can be seen, we instantiated three monitoring tools such as Nagios [245], Google Analytics [252] and a custom-made Events Monitoring component. All of them implement different measures instruments (e.g. Nagios provides several APIs for monitoring different components of systems, services and applications). To this respect, we have wrapped as a service the APIs corresponding to the CPU load and memory usage from Nagios, event tracker from Google Analytics, and HTML DOM events from Events Monitoring. Hence, the monitors are able to collect data from the following channels: usage of CPU and RAM from a system of the IT infrastructure, and user's events from webpages. Finally, although the data collected is also stored in Kafka as the previous use case, the interface of the repository component provides the required schema for instantiating other repositories.

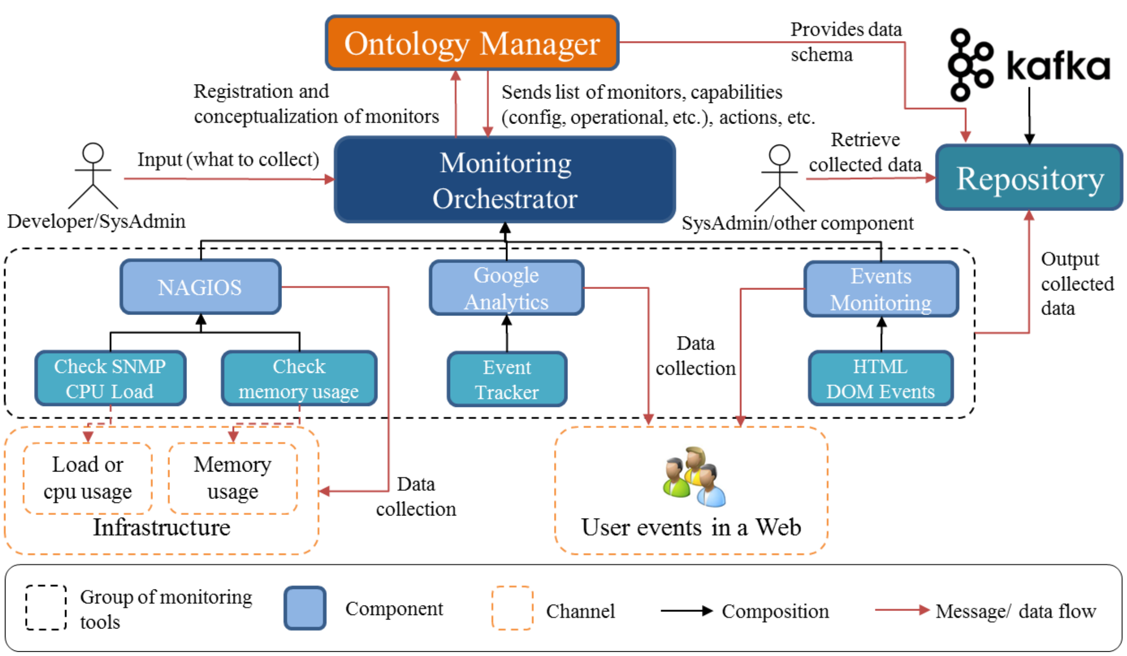


Fig. 45 Monitoring architecture: the IT infrastructure use case

4.5 Implementation of CAMA

A prototype of the framework has been implemented in Java8 with RESTful services built on JAX-RS and running under Apache Tomcat. The code, continuously and iteratively developed, is available at https://github.com/supersede-project/monitor_feedback/tree/socialNetworkMonitoring/monitors.

The framework is embedded as a component of the platform developed in the SUPERSEDE project, but due to its low coupling and SOA architecture, it can also be used standalone.

The monitors currently implemented are:

- Twitter: monitors at real-time the tweets that satisfy a specific search criteria (e.g. keywords, accounts, etc.) and retrieves the messages and some metadata (e.g. user, timestamp, tweetID, etc.). To obtain the tweets, this component uses the Streaming API provided by Twitter[18].
- GooglePlay: monitors at real-time the feedback provided by users to a specific app in GooglePlay and retrieves the messages, ratings and some metadata (e.g. user, timestamp, etc.). To obtain the feedback, this component uses the Google Play Developer API[19].
- AppStore: monitors at real-time the feedback provided by users to a specific app in AppStore and retrieves the messages, ratings and some metadata (e.g. user, timestamp, etc.). To obtain the feedback, this component uses the RSS Feed Generator of iTunes[20].
- HTML clickstreams: monitors the clickstream of a user in an HTML web page. To obtain the clickstream, this component uses JavaScript functions that listen the user events.

Regarding the implementation of the ontology, each level and module of the ontology, and the social networks monitoring tools domain has been implemented separately into the Protégé editor in OWL to facilitate the reuse of the proposed resources. The implemented resources are available at https://github.com/ocabgit/Three-LevelContextOntology.git.

4.6 Validation of CAMA

To validate our proposal we have conducted a set of activities that evaluates how CAMA satisfies the different RQs defined. In particular, we aim to assess that CAMA meets the validation objectives as shown in Table 29.

Table 29. List of validation objectives

RQ#	Validation objective
RQ3.1.1	Assess that CAMA provides the capability to add new monitors with minimal effort.
RQ3.1.2	Assess that CAMA provides the capability to reconfigure the monitors automatically.
RQ3.1.3	Assess that CAMA provides the capability to replace a failing monitor automatically.
RQ3.2	Assess that CAMA provides the capability to access the monitored data using a rich semantics method following an open-world assumption.

4.6.1 Validation of RQ3.1.1

In this subsection, we describe the validation that assesses that CAMA provides the capability to add new monitors with minimal effort. The addition of new monitors do not require any change on the code, and only requires to register it with its inputs and outputs into the ontology of CAMA.

[18] https://dev.twitter.com/streaming/overview
[19] https://developers.google.com/android-publisher/
[20] https://rss.itunes.apple.com/

To this aim, we have prepared an exercise to extend the ontology with a new monitor that was filled by different practitioners. We assume that the new monitor to be added already provides a web service interface, and therefore, there is no need to programmatically implement a web service wrapper (although, it could be done if needed).

4.6.1.1 Protocol of the validation

To carry out this validation, we conducted the following tasks:

1) Definition of a scenario/exercise consisting of the development of a domain ontology for conceptualizing monitoring tools where inputs, outputs and general capabilities should be specified. The scenario was beta-tested by 2 researchers of our group whose feedback helped to make the instrument fit for purpose. Details of the scenario/exercise are provided at [253].
2) Selection of the participants to run the exercise. To conduct this task we recruited 18 researchers and practitioners with different expertise and background on monitoring of services and applications, and/or ontologies.
3) Explanation to the participants of the aim of this practice, providing them the supporting material described in Section 3.3.3.1 of Chapter 3, and also giving them general considerations of the scenario.
4) Execution of the exercise by the participants and evaluation of the time taken for them to build the ontology and the quality of the result.

4.6.1.2 Results of the validation

The results of the 18 exercises are presented in Table 30 according to the following criteria:

- **Level-Ont.** Level of expertise in modelling ontologies, from 1 (low) to 5 (high).
- **Level-Mon.** Level of expertise in monitoring systems, from 1 (low) to 5 (high).
- **T1.** Time it takes to read and understand the exercise and supporting material.
- **T2.** Time it takes to elaborate and write down the solution.
- **Usability**. Subjective opinion given by the participant stating whether the proposed ontology is usable or not to conceptualize any monitoring tool.
- **Correctness.** Quality of the model designed by the participant with possible values:
  - **Excellent**. All needed classes and properties that maintain the model consistent were specified.
  - **Very good**. All needed classes were specified; however, some property (data or object) was not specified.
  - **Regular**. The model is not entirely wrong, but several inconsistencies were found.
  - **Bad**. The model does not represent the classes and properties that we expected, and several inconsistencies were found.
- **Improvements.** It represents whether the participant considers that the abstract layer of the proposed ontology (upper and middle level classes) should be improved to represent any kind of monitoring tool.
- **Difficulties.** It represents the difficulties of the participant to understand the exercise and supporting material.

Table 30. Time consuming for building domain ontologies

Level-Ont	Level-Mon	T1	T2	Usability	Correctness	Improvements	Difficulties
4.5	1	00:06:30	00:14:13	Yes	Very good	No	Yes
4	1	00:05:00	00:18:00	Yes	Very good	No	No
4	3	00:08:00	00:15:00	Yes	Excellent	No	No
3	3	00:05:35	00:15:32	Yes	Very good	No	No

3	1	00:10:07	00:11:56	Yes	Very good	Yes	No
3	1	00:11:00	00:22:00	Yes	Very good	Yes	No
3	2	00:04:12	00:09:19	Yes	Excellent	Yes	No
3	2	00:07:45	00:14:55	Yes	Excellent	Yes	Yes
3	4	00:08:10	00:15:00	Yes	Excellent	No	No
3	3	00:04:00	00:11:00	Yes	Excellent	No	No
2	3	00:04:00	00:10:00	Yes	Very good	No	No
2	3	00:14:45	00:18:36	Yes	Very good	No	No
2	1	00:07:37	00:30:22	Yes	Regular	No	No
2	5	00:07:56	00:12:29	Yes	Excellent	No	Yes
1	1	00:15:21	00:14:00	Yes	Very good	No	Yes
1	2	00:13:00	00:17:00	Yes	Very good	No	No
1	3	00:15:31	00:10:53	Yes	Excellent	No	No
1	1	00:03:40	00:14:21	Yes	Very good	No	Yes

The analysis and evaluation of the results obtained from the empirical study are described and classified as follows considering the more relevant findings of Table 30:

- **Level-Ont vs (T1, T2 and difficulties).** To identify some correlation between these criteria, first we grouped the participants by level of modelling ontologies, i.e., participants belonging to low level of modelling ontologies (level 1), middle level (level 2), middle-high level (level 3) and high level (level 4 to 5). Second, we calculated the average time taken by the participants of each group for understanding the exercise and supporting material (T1). The results obtained show a clear correlation between these two criteria since we found that the average time taken from group level 1 (avg. 00:11:53) to group level 4 (00:06:30) was considerably decreasing (see the tendency line in Fig. 46). Similarly, although there also exists a decreasing correlation between the level of expertise in modelling ontologies of the groups and T2, the slope of the tendency line depicted in Fig. 47 is much more moderated. Only the participants belonging to group level 1 have used a bit more of time than those belonging to level 2, 3 and 4. To understand these findings, we looked at the answers given in the *difficulties* found, and although most of them didn´t find difficulties, we found that most of the participants (level 1 and 2) took time for understanding specific details of each of the elements specified in the documentation. Such situation was decreasing in participants belonging to level 3 and 4. Finally, these results allow us to conclude that the effort to model the required ontology is not excessively time consuming in T1 nor T2. It only takes on average 10:14 minutes (T1) and 16:37 (T2) for beginners and intermediate participants (level 1 and 2), and 07:02 minutes (T1) and 13:34 minutes (T2) for more advanced participants (level 3 and 4).

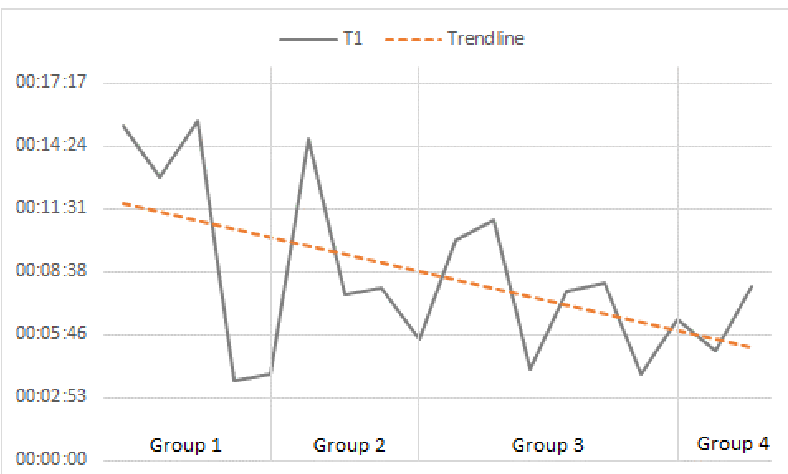


Fig. 46 Correlation between T1 and groups level-ont

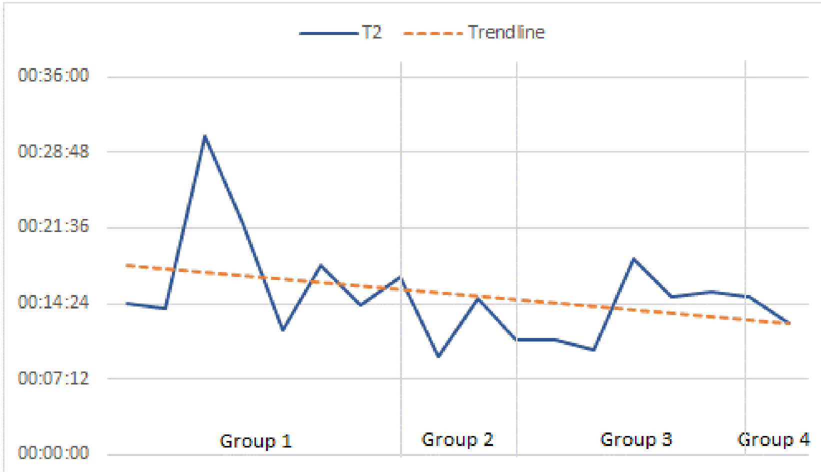


Fig. 47 Correlation between T2 and groups level-ont

- **Level-Mon vs (T1, T2 and difficulties).** The most important finding in this case was that the level of expertise in monitoring systems was not an important variable that affected the process for reusing the proposed ontology and for modelling the required extension to conceptualize marketplace monitoring tools. According to the answers of the participants, this finding is because the ontology is easy to understand and intuitive, and the supporting material is clear and helpful to understand the main terms of the model. Hence, any user without having expertise in monitoring systems can immediately start using the ontology.
- **Correctness vs (Difficulties, level-ont and level-mon).** We also evaluated the correlation between the quality level of the model designed by the participants and the *difficulties* that they found while reading the exercise and supporting material. As it can be seen in Table 30, only 28% of the participants found difficulties when reading the exercise and supporting material, however these difficulties have not affected the quality of their model since all of them oscillate between very good and excellent in the *correctness* results depicted in the table. Hence, no correlation was found between these two criteria. Additionally, as it can be seen in the *correctness* criterion, the level of expertise in modelling ontologies (*level-ont*) and monitoring systems (*level-mon*) was not an important factor to decrease the correctness of the models. In fact, only one of the participants with level 2 in modelling ontologies and 1 in monitoring expertise provided a solution to the proposed exercise that was of regular quality. Although the model was not entirely wrong, several inconsistencies were found in his solution. These results allow us to conclude that the ontology is easy to extend and the resulting extended ontology is, in most cases, of very high-quality.
- **Usability.** The findings based on the results of *usability* show that 100% of the participants found the proposed ontology useful, easily extensible and reuse that provides the concepts and relations needed to conceptualize any monitoring tool. We consider that this satisfactory result is due to the different features offered by the proposed model highlighting the intuitive and easy representation of entities and context information to characterize a monitoring tool, the precise and concise primitives needed to represent a monitoring tool, the short size that makes it easy to handle, etc.
- **Improvements.** The results of *improvements* show that 22% of the participants provided a suggestion to improve the proposed model. To this respect, the suggestions of the participants were related to adding more attributes (i.e. data type properties) to describe this type of monitoring tools. This suggestion does not affect the proposed ontology since it depends on the scenario and the modeler to include more or less attributes to describe each domain class specified. In this case, the attributes that we proposed represent the basic ones to describe a type of monitoring tool. Another suggestion from the participants was the creation of a middle-level class named "MarketPlaceServices" as a subclass of "CompService" since they do not consider a marketplace

service a type of “SocialNetworkingServices”. We consider that such suggestion manifest the reusability capabilities of the proposed ontology to model any type of monitoring tool. This is because the semantic represented in the ontology has been taken as a basis to model a new class that rather than modify the proposal it represents an extension. It is worth noting that we consider correct both solutions (the proposed model and the suggested improvement) as long as this differentiation remain consistent with the other classes of the model.

4.6.2 Validation of RQ3.1.2

In this subsection, we describe the validation that assesses that CAMA provides the capability to reconfigure the monitors automatically. It is worth to remark that the analysis and decision making process that would trigger the adaptation is out of scope of this thesis. Here we validate how CAMA enacts a reconfiguration, assuming that an external component has already conducted an analysis and decision making process that has resulted in a reconfiguration need.

4.6.2.1 Protocol of the validation

The protocol of this validation is as follows:

1) The monitors to reconfigure will be the monitors of Twitter, GooglePlay and AppStore. The monitor of HTML clickstreams will not be tested, as in its current implementation; this monitor does not provide any parameter to reconfigure.
2) For each monitor, we will execute at runtime 10 different reconfigurations. Each reconfiguration will be requested just after the previous reconfiguration has been executed.
3) Each reconfiguration will change all the parameters that can be reconfigured for that monitor

For each reconfiguration, we will measure:

1) The success or failure of the reconfiguration
2) The time to complete the reconfiguration

4.6.2.2 Results of the validation

Here we present the results of the validation following the protocol described previously.

Table 31. Results of monitoring reconfiguration activities

Monitor	# executions	Success rate	Avg. time to reconfigure	Attributes involved in the reconfiguration
Twitter	10	10/10	0,141s	Time slot, keyword expression
GooglePlay	10	10/10	0,089 s	Time slot, App Id
AppStore	10	10/10	0,094 s	Time slot, App Id

As shown in Table 31, all monitors have been reconfigured with a 100% success rate. The average time to reconfigure the monitors of Twitter, GooglePlay and AppStore have been 0,141s, 0,089s and 0,094, respectively.

4.6.3 Validation of RQ3.1.3

In this subsection, we describe the validation that assesses that CAMA provides the capability to replace a failing monitor automatically. It is worth to remark that the analysis on whether a monitor is failing or not is out of scope of this thesis. Here we validate how CAMA switches from one monitor to another, assuming that an external component has detected a failing monitor.

4.6.3.1 Protocol of the validation

The protocol of this validation is as follows:

1) The monitors used to replace their monitoring tools will be the monitors of GooglePlay and AppStore. At the current state of implementation, these 2 monitors are the only ones that have more than one monitoring tool.
2) For each monitor, we will start with a specific monitoring tool. We will simulate that after 1 minute, the used monitoring tool fails and we will trigger CAMA to replace the used monitoring tool for another.
3) We will execute the aforementioned process 10 times for each monitor.

For each monitoring tool replacement, we will measure:

1) The success or failure of the monitoring tool replacement
2) The time to complete the monitoring tool replacement.

4.6.3.2 Results of the validation

Here we present the results of the validation following the protocol described previously.

Table 32. Results of replacing monitoring tools

Monitor	# executions	Success rate	Avg. time to replace monitoring tool	Monitoring tools replaced
GooglePlay	10	10/10	0,081 s	GooglePlayAPI ↔ AppTweak-googlePlay
AppStore	10	10/10	0,196 s	ITunesApple ↔ AppTweak-appStore

As shown in Table 32, all monitoring tools have been replaced for another one providing the same functionality with a 100% success rate. The average time to replace one monitoring tool for another has been 0,081s for GooglePlay and 0,196s for AppStore.

4.6.4 Validation of RQ3.2

In this subsection, we describe the validation that assesses that CAMA provides the capability to access the monitored data using a rich semantics method following an open-world assumption. To do so, we need to validate: (1) The consistency of the ontology, and (2) the reasoning capabilities of the ontology.

4.6.4.1 Validation of the consistency of the ontology

In this case, we want to validate that the knowledge represented in the ontology is consistent with its scope, i.e., that the ontology has information enough to answer the competence questions related to the management of monitoring tools (see Section 3.3.3 of Chapter 3). The expected results should demonstrate that layers and modules, the extension of the ontology, and the conceptualization of the domain ontology that manages the social networks monitoring tools of CAMA (see Fig. 17) are correct and consistent. Note that we have made some queries related to this validation in the previous chapter (see Section 3.4.2) that are complemented here.

4.6.4.1.1 Protocol of the validation

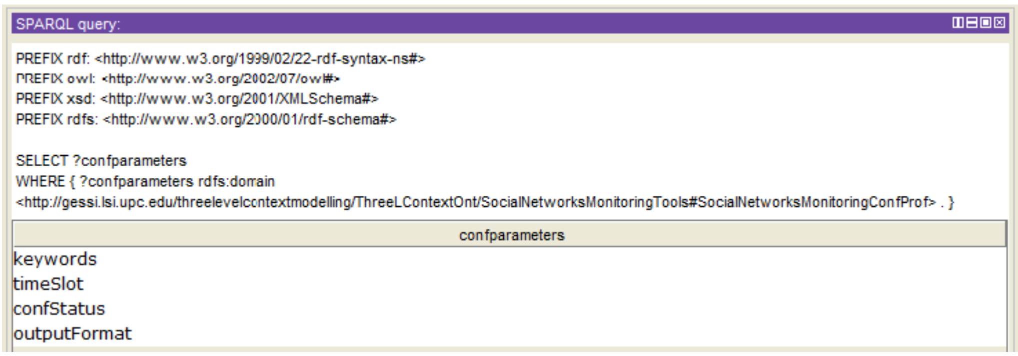
To carry out this validation, we conducted the following tasks:

1) Translation of the following competence questions a) what are the parameters that can be configured in a monitoring tool? b) what are the inputs, outputs and configurations capabilities of monitoring tools? into the semantics of the SPARQL query language.
2) Execution of the queries into the Protégé editor.

4.6.4.1.2 Results of the validation

The queries and they related results are specified in the following figures.

Fig. 48 shows a query that can retrieve all the configurable parameters in social networks monitoring tools. As it can be seen, there are four common parameters that can be configured in monitoring tools for social networks.



SPARQL query:

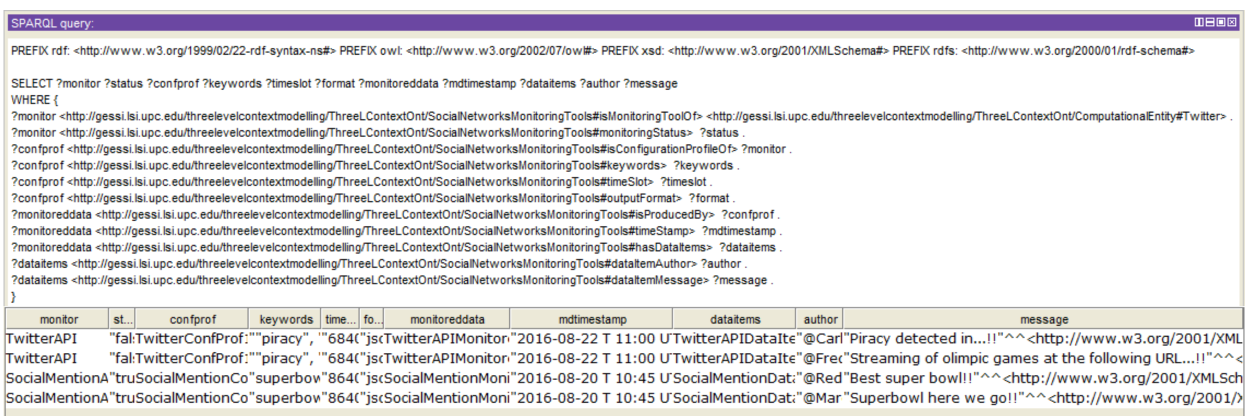
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?confparameters
WHERE { ?confparameters rdfs:domain
<http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#SocialNetworksMonitoringConfProf> . }
```

confparameters
keywords
timeSlot
confStatus
outputFormat

Fig. 48 SPARQL query of parameters that can be configured in a social network monitoring

Fig. 49 is depicted a detailed query to obtain monitoring tools, configuration instances and their respective values that describe the inputs, outputs and configuration capabilities of social networks monitoring tools. As it can be seen, there are two monitoring tools, one of them with status true, i.e., active to supervise tweets. Each monitoring tool is related to configuration profiles that produce an output, i.e., monitored data (e.g., author and message).



SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?monitor ?status ?confprof ?keywords ?timeslot ?format ?monitoreddata ?mdtimestamp ?dataitems ?author ?message
WHERE {
?monitor <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#isMonitoringToolOf> <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/ComputationalEntity#Twitter> .
?monitor <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#monitoringStatus> ?status .
?confprof <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#isConfigurationProfileOf> ?monitor .
?confprof <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#keywords> ?keywords .
?confprof <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#timeSlot> ?timeslot .
?confprof <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#outputFormat> ?format .
?monitoreddata <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#isProducedBy> ?confprof .
?monitoreddata <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#timeStamp> ?mdtimestamp .
?monitoreddata <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#hasDataItems> ?dataitems .
?dataitems <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#dataItemAuthor> ?author .
?dataitems <http://gessi.lsi.upc.edu/threelevelcontextmodelling/ThreeLContextOnt/SocialNetworksMonitoringTools#dataItemMessage> ?message .
}
```

monitor	st...	confprof	keywords	time...	fo...	monitoreddata	mdtimestamp	dataitems	author	message
TwitterAPI	"fal:	TwitterConfProf:	""piracy",	'"684(	"js(	TwitterAPIMonitor	"2016-08-22 T 11:00 U	TwitterAPIDataIte	"@Carl	"Piracy detected in...!!"^^<http://www.w3.org/2001/XML
TwitterAPI	"fal:	TwitterConfProf:	""piracy",	'"684(	"js(	TwitterAPIMonitor	"2016-08-22 T 11:00 U	TwitterAPIDataIte	"@Frec	"Streaming of olimpic games at the following URL...!!"^^<
SocialMentionA	"tru	SocialMentionCo	"superbow	"864(	"js(	SocialMentionMoni	"2016-08-20 T 10:45 U	SocialMentionDat:	"@Red	"Best super bowl!!"^^<http://www.w3.org/2001/XMLSch
SocialMentionA	"tru	SocialMentionCo	"superbow	"864(	"js(	SocialMentionMoni	"2016-08-20 T 10:45 U	SocialMentionDat:	"@Mar	"Superbowl here we go!!"^^<http://www.w3.org/2001/>

Fig. 49 SPARQL query of inputs and outputs of social networks monitoring tools

4.6.4.2 Validation of the reasoning capabilities of the ontology

The reasoning capabilities of an ontology is a key value that can also be exploited in our proposal. Ontologies of monitoring tools as the depicted in Fig. 17 that manages social network monitoring tools of CAMA have the benefit of supporting reasoning with the purpose of deducing new knowledge based on the ontology and rule based reasoning that can be useful for automatic configurations of monitoring tools.

4.6.4.2.1 Protocol of the validation

For demonstrating the reasoning process on the proposed ontology, we consider the following reasoning capabilities:

1) The ontology-based reasoning that uses the existing reasoning rules already defined in the semantics of OWL (e.g., rdfs:subClassOf, owl:sameAs, etc.).
2) The rule-based reasoning that is not included in the semantics of OWL and therefore, the rules should be explicitly defined by users.

4.6.4.2.2 Results of the validation

For illustrating the ontology-based reasoning, the transitivity rule of the *isAffectedBy* property infers the following situational context of a monitoring tool: if an instance of the *SocialNetworksMonitoring* class is affected by the server overload caused by several requests to it and this sever overload is also affected by the server location, then the instances of the *SocialNetworksMonitoring* are also affected by the server location (see Table 33).

Table 33. Ontology reasoning for deriving high-level context of monitoring tools

Explicit context	***Implicit context***
<owl:ObjectProperty rdf:ID="isAffectedBy"> <rdf:type="owlTransitiveProperty"/> </owl:ObjectProperty> <SocialNetworksMonitoring rdf:ID="SocialMentionAPI"> <isAffectedBy rdf:resource="#Server Overload"/> </SocialNetworksMonitoring> <States&Status rdf:ID=" ServerOverload"> <isAffectedBy rdf:resource="#ServerLocation"/> </States&Status>	<SocialNetworksMonitoring rdf:ID="SocialMentionAPI "> <isAffectedBy rdf:resource="#ServerLocation "/> </SocialNetworksMonitoring>

For illustrating the rule-based reasoning, consider the following rules defined using the semantic of Protégé:

1. *Orchestrator(?a), SocialNetworksMonitoring(?b), SocialNetworksMonitoringConfProf(?c), Social NetworksMonitoredData(?d), SocialNetworksDataItems(?e), hasConfigurationProfile(?b, ?c), perfor msActionsOn(?a, ?c), isProducedBy(?d, ?c), hasDataItems(?d, ?e), hasMessage(?e, Piracy) -> reducesTimeSlotOf(?a, ?c)*
2. *Orchestrator(?a), SocialNetworksMonitoring(?b), hasMonitoringTools(?a, ?b), monitoringOf(?b, Twitter), hasSatus(?b, Off) -> changesStatesStatusOf (?a, ?b)*

These rules infer the activities of the orchestrator over the monitoring tools and configuration models (see Fig. 42), if all the statements are true, i.e., if all the context information retrieved match with the rule, then it is deduced an activity. For instance, in the first rule is specified that if a specific keyword (e.g., “Piracy”) is detected in a Twitter message, then the orchestrator should change the timing of the instance associated to retrieve the most updated results. The structure of the same rule can also be applied for changing the monitoring tool used, e.g., if the data items retrieved by a specific instance of the *SocialNetworksMonitoringConfProf* class are null, then change the parameters of the instance or change the monitoring tool used. This rule can also be useful when a monitor is working but the results retrieved are null because of delays of the server. Similarly, the second rule can be useful when the orchestrator receives a request for monitoring tweets. In this case, if the orchestrator has registered Twitter monitoring tools and these monitoring tools have status “Off”, then the orchestrator should change the status of the instances associated to these types of monitors to “On”.

“Intelligence is the source of technology. If we can use technology to improve intelligence, that closes the loop and potentially creates a positive feedback cycle”

- Eliezer Yudkowsky

CHAPTER 5

Supporting the context life cycle: closing the loop

Context-aware computing is key in the development of new services and applications that are increasingly requiring being aware of their context and that will impact social inclusion for the emerging information society [254]. For this purpose, the context life cycle plays an important role in context-aware computing. In this regard, the main contribution of this chapter of the thesis is focused on providing a holistic view of the main results of the thesis namely the three-level context ontology (3LConOnt) and the context-aware monitoring architecture (CAMA) with the aim of closing the loop that supports the context life cycle. For such purpose, we propose a generic context-aware framework that can be able to allow the execution of different types of context-aware scenarios.

To support the context life cycle the proposed framework should provide the elements and relations that can fulfill the phases of the cycle (see Fig. 50). Hence, the context-aware framework integrates the previous proposals of the thesis as follows: for context modelling, context reasoning and high-level context dissemination it integrates the three-level context ontology (3LConOnt) that was proposed and described in Chapter 3. For context reasoning and low-level context dissemination, it integrates the context-aware monitoring architecture (CAMA) that was proposed and described in Chapter 4.

Fig. 50 Closing the context life cycle

5.1 The proposed context-aware framework

The main result of this contribution of the thesis is a generic context-aware framework that is designed around three main components namely monitoring infrastructure, manager engine and context ontology (see Fig. 51). The figure shows the role that can play the three-level context ontology and the context-aware monitoring architecture in different use cases of the service-oriented computing. This includes: structuring and unifying context data; configuration, adaptation and evolution of a monitoring infrastructure; ranking and adaptation of services and applications; evolution and adaptation of personalized software; improving the QoE of the user; etc. The context-aware framework is intended to support the context life cycle taking as kernel the proposed context ontology.

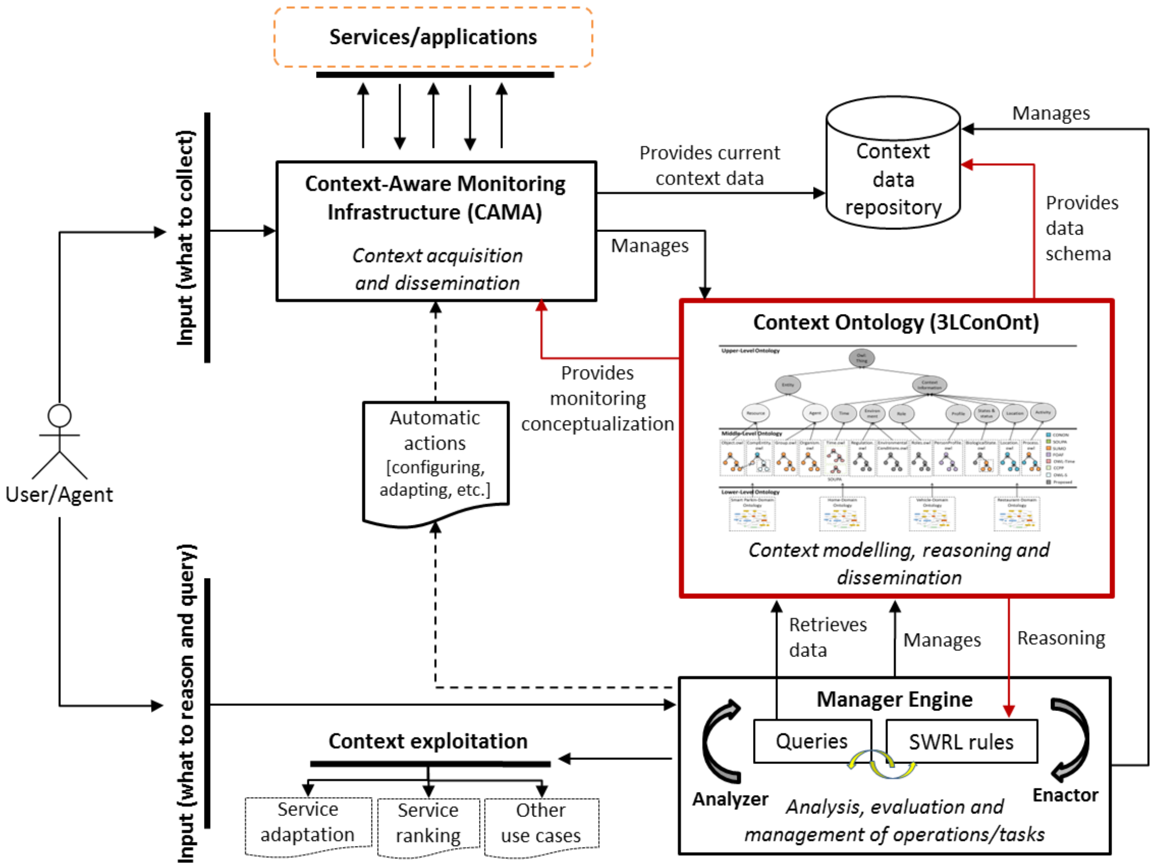


Fig. 51 Context life cycle based on the results of this PhD thesis

As it can be seen in the figure, the context ontology proposed in this thesis is responsible for structuring, reasoning and disseminating high-level context data. Thus, the model provides the data schema of the context data repository to unify and structure the acquired current context data coming from the monitoring infrastructure component. The manager engine that has two important roles namely analyser and enactor manages the context ontology and the context data repository to perform its tasks. First, the analysis and evaluation of the context data sent from the context-aware monitoring infrastructure. Second, based on the analysis and evaluation performed, the manager engine can decide and enact the needed actions in an application domain (e.g., supporting the configuration/adaptation of the monitoring infrastructure and other services or applications). Since queries and rules depend on the application domain, the user/agent plays an important role to specify them based on the functional requirements (e.g., competency questions) of a domain ontology. Hence, the manager engine can be able to extract direct or indirect information from the context ontology that provides new facts through its reasoning capabilities.

Consider the following case of social networks monitoring tools. Once the manager engine has detected adaptation events through the internal loop of queries and reasoning capabilities of the context ontology, it is ready for triggering adaptations and therefore, it evaluates what should be adapted. Hence, the manager engine manages such adaptations and generates a JSON file (see Fig. 52) that should indicate the adaptations of the monitoring system. Finally, such file is sent to the orchestrator of CAMA to orchestrate the adaptations and indicate to the ontology manager (see Fig. 42), the changes that should be updated in the context ontology.


```
{
    "timeStamp": "2016-10-20T20:10:30:201",
    "SocialNetworksMonitoringConfProf":{
         "timeSlot": "time in seconds",
         "repositoryEndpoint": "URL",
         "repositoryTopic": "topic",
         "keywordExpression": "(keyword1 OR keyword2) AND (keyword 3
OR keyword4 OR keyword5)",
         "accounts": ["account1", "account2", "account3"],
         "state": "ON/OFF",
         "id": "id configuration"
      }
}
```


Fig. 52 JSON file for adapting social networks monitoring tools

We also consider that the proposed context-aware framework is generic enough to be aligned with other context-aware perspectives. For instance, the awareness layered view suggested in [255] can be processed and mapped in our proposal as follows: 1) the context middleware layer can be addressed by the interaction of the motoring infrastructure with the context ontology, since these components collect and maintain the context data; and 2) the awareness and sensitivity layer can be addressed by the interaction of the manager engine and the context ontology, since these components can assess and classify the situational context of an entity as well as select, decide and execute the feasible actions that can be enacted in a context-aware service or application.

5.2 Validating the framework

To validate the framework, we have prepared a running example based on the reconfiguration of a monitor for Twitter. The purpose is demonstrating that a proper configuration of the monitor, through a set of adaptation rules, improve its behaviour with respect to some quality criteria.

5.2.1 Validation set-up

We have instantiated a self-reconfigurable monitor of Twitter to monitor the tweets related to the World Chess Championship 2016. The World Chess Championship 2016 was a series of chess matches between the world champion Magnus Carlsen and the challenger Sergey Karjakin to determine the new World Chess Champion. We have monitored the tweets related to the Championship during the last four tie-breaker games celebrated in 30th November (won by Carlsen, by the way). The tweets monitored can then be used by 3rd parties as a source for sentiment analysis (e.g. to identify if the game was exciting, to know which player has more supporters, etc.).

The monitor is configured by means of a keyword expression that defines the tweets that should be gathered at runtime. To this respect, we have designed two different keyword expressions to configure the monitor (see Table 34). The *keywordExpression1* only uses the official hashtags of the event, whereas the *keywordExpression2* includes the keywords of *keywordExpression1* plus the names of the chess players and other relevant keywords related to the event.

Table 34. Keyword expressions to configure the monitor of twitter

ID	Keyword Expression
1	worldchess2016 OR CarlsenKarjakin
2	worldchess2016 OR CarlsenKarjakin OR MagnusCarlsen OR Magnus-Carlsen OR Carlsen OR SergeyKaryakin OR Karjakin OR Sergey-Karjakin OR theworldchess

In this validation, we are interested that the self-reconfigurable monitor alternates between *configuration1* and *configuration2* depending on the following context: when the number of tweets obtained is low, the monitor should focus on obtaining more tweets (in information retrieval, this is called recall [256]). Whereas when the number of tweets obtained is high, the monitor should focus not on getting so many tweets but ensuring that the tweets obtained are relevant to the monitored event (in information retrieval, this is called precision [256]).

For this validation, we consider that an adequate amount of tweets is between 50 and 200 tweets per minute[21]. If we obtain less than 50 tweets per minute, the monitor should be reconfigured to use the *KeywordExpression2* to get more data. On the contrary, if we obtain more than 200 tweets per minute, the monitor should be reconfigured to use the *KeywordExpression1* to obtain less tweets but more relevant. For values between 50 and 200 tweets per minute (i.e. neither too high nor too low), the monitor should keep the configuration it had in the previous iteration to avoid unnecessary reconfigurations.

In this experiment, we have configured the time slot of the monitors to run every 2 minutes. Therefore, the thresholds for reconfiguration would be 100 and 400 respectively. The defined rules in the ontology using SWRL language are shown below:

1. *SocialNetworksMonitoring(?SNM), SocialNetworksMonitoringConfProf(?SNMC P1), SocialNetwo rks MonitoringConfProf(?SNMCP2), SocialNetworksMonitoredDat a(?SNMD), SocialNetworksDa ta Items(?SNDI), hasConfigurationProfile(?SNM, ?S NMCP1), isProducedBy(?SNMD, ?SNMCP 1), has DataItems(?SNMD, ?SNMDI), hasNumDataItems(?SNMD, ?NumTweets), lessThan(?Num Tweets, 100) -> hasConfigurationProfile(?SNM, ?SNMCP2)*
2. *SocialNetworksMonitoring(?SNM), SocialNetworksMonitoringConfProf(?SNMC P1), SocialNetwo rks MonitoringConfProf(?SNMCP2), SocialNetworksMonitoredDat a(?SNMD), SocialNetworksDa ta Items(?SNDI), hasConfigurationProfile(?SNM, ?S NMCP2), isProducedBy(?SNMD, ?SNMCP 2), has DataItems(?SNMD, ?SNMDI), hasNumDataItems(?SNMD, ?NumTweets), greaterThan(? NumTweets, 400) -> hasConfigurationProfile(?SNM, ?SNMCP1)*

We ran the monitor for 10 hours: just before (18:40 - 20:00 GMT+1), during (20:00 - 00:50) and after (00:50 - 04:50) the tie-breaker games. In the next subsection, we analyze and compare the results of three different configurations: *configuration 1* fixed, *configuration* 2 fixed, and *self-reconfigurable* changing between one and the other according to the rules above. For that, we used multiple instances of the monitor running in parallel.

5.2.2 Results of the validation

In this section, we describe the results obtained after executing the experiment presented above. Using *configuration 1*, we obtained a total of 15978 tweets; using *configuration 2*, we obtained a total of 34962 tweets; and using the self-adaptive strategy (*self-reconfigurable*), we obtained a total of 29652 tweets. Fig. 53 depicts the number of tweets obtained in the different configurations distributed along time.

[21] Those thresholds have been defined for validation purposes in order to show how the monitor can be reconfigured at runtime. Defining thresholds that are justified by statistical or formal methods are out of the scope of the thesis

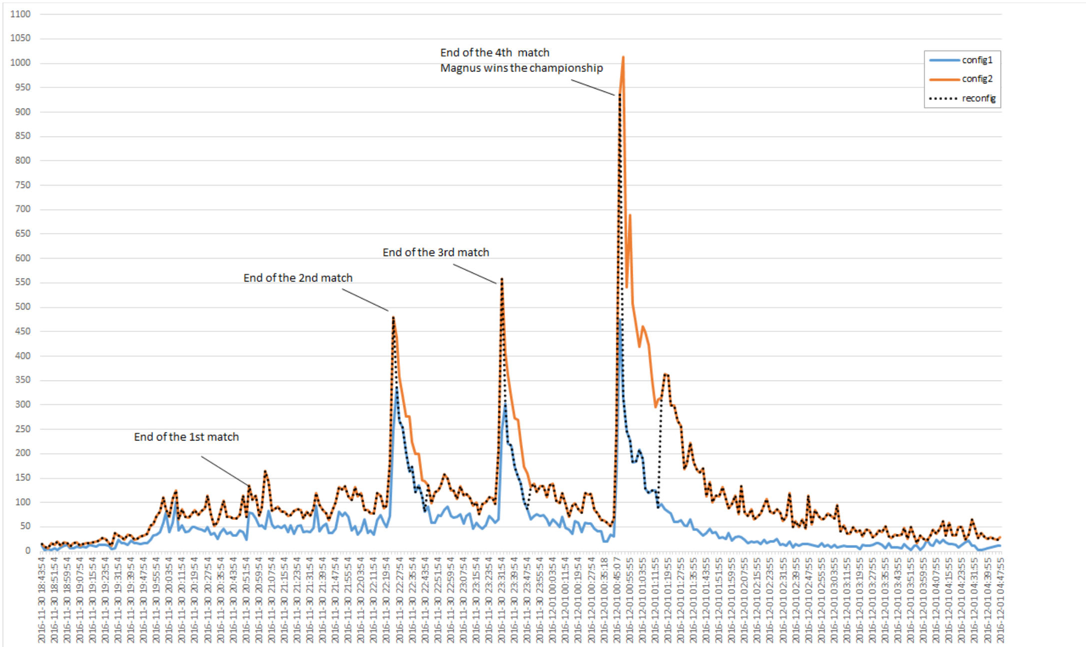


Fig. 53 Tweets monitored during the tie-breaker games between Magnus Carlsen and Sergey Karjakin

As shown, initially the number of tweets was very low, and therefore the *self-reconfigurable* configuration used *configuration 2* to get as much tweets as possible. Then, at 22.23h (GMT+1), the number of tweets reached more than 400 tweets and the monitor was reconfigured to use *configuration 1* to reduce the number of tweets and ensure that the tweets obtained were relevant. 20 minutes later, at 22.43, the number of tweets was lower than 100 tweets, and it was reconfigured again. During the experiment, the monitor was reconfigured 6 times. The complete list of reconfigurations done is shown in . The time to conduct these reconfigurations is negligible, as it takes between 100 and 200 ms in an Intel core i7 at 2,4 Ghz.

Table 35. List of reconfigurations done

Time	Condition	Value (#tweets)	Action
22.23	uses config2 AND numTweets >400	480	config2 → config1
22.43	uses config1 AND numTweets <100	82	config1 → config2
23.31	uses config2 AND numTweets >400	559	config2 → config1
23.47	uses config1 AND numTweets <100	86	config1 → config2
00.45	uses config2 AND numTweets >400	936	config2 → config1
1:13	uses config1 AND numTweets <100	90	config1 → config2

"Every new beginning comes from some other beginning's end"

- Seneca

CHAPTER 6

Conclusions

Maintaining adequate service levels is crucial for the comfort of users and customers. In this regard, different aspects are responsible for maintaining the required levels and to avoid risks, being context-awareness one of the most important aspects to be considered in the health of services and other entities that conform the value structure.

In the present PhD thesis, we contribute to context-awareness in the scope of service-oriented computing by supporting the context life cycle through an improved and consolidated context model for context reasoning and high level context dissemination, and a context-aware monitoring architecture for context acquisition and low level context dissemination. For this purpose, the work of the thesis is focused on 1) a state of the art of existing contributions of context modelling based on ontologies in order to identify some gaps and patterns presented in the current context modelling; 2) the development of a three-level context ontology with the aim of solving the lacks identified in the state of the art; and 3) the development of a context-aware monitoring architecture for solving lacks also identified through a state of the art of existing context-aware architectures. In summary, this PhD thesis faces the previous contributions by exploring and answering three big research questions as follows:

RQ1) What to acquire and to model? For solving this question, we conducted a systematic mapping study on context modeling based on ontologies focused on analyzing and evaluating pieces of context knowledge including classes (context information and entities), properties and functions. The survey consolidates the information retrieved and provides the necessary background to build or reuse context models. For this purpose, we designed and followed a review protocol that has allowed gathering 138 proposals to answer the research questions that we define as part of this process.

We consider that the results of the review can be useful in the development of future context models more consolidated and in some way more standardized by considering all the semantic factors and completeness of the context resources so far provided such as classes of context information and entities, datatype and object properties, etc. making available a model easy to reuse in any contextual domain.

The main reasons for formulating a consistent proposal given the perspective and findings of the study on context modelling are summarized as follows:

1) *Clarity*: compiling all the semantic definitions and representations of the primitives proposed in the analyzed contributions, we identified what so hard can be finding equivalences among these primitives. Some of them are too brief or too abstract, so much that, delimit their meaning and representation in other models. At this respect, we consider that the context-aware characterization of a service needs the capture of a clear and concise context model easy to be comprehensive.
2) *Generality*: most of the analyzed contributions were focused on the needs of an application domain providing domain-dependent ontologies and primitives hardly to reuse in new domains. At this respect, we consider that a model should provide the feasibility and facility for adding new context knowledge. Furthermore, the development of context-aware services needs the capture of the

generic view of a context model (upper or middle level ontologies) for the creation of prospective services.

3) *Uniformity*: most of the analyzed contributions proposed ontologies without mappings to, and equivalences with, a foundational ontology, decreasing thus the semantic quality, coherent vocabulary, reusability and interoperability of proposed models. The reasons are the time consuming and operational cost when finding a common knowledge and getting acquainted with foundational ontologies. However, we consider that these costs can be balanced by increasing the sharing of knowledge and the interoperability between context-aware services when the model is grounded by foundational ontologies;
4) *Explicit representability*: we identified that the ontologies that define primitives (concepts and properties) without an explicit representation between these primitives and their related semantics (definitions, scope and specific relationships) considerably reduce their reasoning capabilities and therefore, also their reusability and capability of sharing knowledge between services. Hence, we consider that a model should explicitly define and represent such primitives and semantics.
5) *Ambiguity*: the study showed the existence of a set of inconsistencies in a model (e.g., classes also specified as properties and instances). Such issue can be a general worry of developers for reusing inconsistent models, since its main effect is also over the reasoning capabilities of the ontology. For instance, due to the noise inserted in a context model, the assertions by reasoning can produce different inconsistencies (e.g., an entity can be located in two places, a service can be asserted also as a project, etc.). Hence, we consider that the models should be subjected to different validations such as satisfying competence questions, identify inconsistencies by means of automatic tools, etc.
6) *Modularity*: the findings of the study showed that the ontologies with a modular perspective were the most ones reused by other ontologies (e.g., CONON and SOUPA). It seems that putting all the concepts together without boundaries, make barriers emerge and increases the likelihood of not reusing an ontology. Given the modularity of the proposal, the developers can reuse only the modules in which they are interested and therefore, reduce the processing time and operational costs of the ontology.

We believe that this panoramic view on the anatomy of context models may be a good reference for prospective researchers and practitioners on the field, especially with the aim of avoiding new classifications or definitions that may be contrary to the established practices that we have found.

RQ2) How to model? For solving this question, we designed a three-level context ontology (3LConOnt) for context modelling to overcome some drawbacks identified when answering RQ1, particularly related to reusability, extensibility and adaptability issues, and the lack of unified and consolidated context knowledge. To solve such gaps of context modelling, the proposed ontology adopts a level-based modelling strategy comprising upper, middle and lower levels of abstraction:

- The upper-level provides high level classes that have been consolidated from the context models studied in the systematic mapping. The aim of this upper-level is to provide a basic taxonomy of context classes that are independent of any particular problem or domain, and that can represent very general context concepts like time, location, agent, etc. Any context model should appraise this taxonomy in order to prevent conceptual and terminological ambiguities.
- The middle-level represents a bridge between the other two levels, and provides an easy way to reuse and to extend ontological resources of existing context models from a modular perspective yielding a clear schema of context knowledge reutilization. The main contribution at this level has been the effort of analyzing, selecting and combining many useful vocabularies from different context models that were unified into modules that better suit the patterns found in existing proposals. For this purpose, we did not import any ontology directly but rather we gathered parts of different ontologies to be integrated into modules.
- The lower-level represents a set of detailed classes highly dependent on the domain. The aim at this level is that domain-specific ontologies proposed in existing contributions or developed from

scratch can be defined and structured by extending the appropriate classes of the modules specified in the upper and middle-level. At this level, we represented different domain ontologies for validating some aspects of the proposed context model.

From this perspective, the three-level context ontology allows the conceptualization and management of new or existing context knowledge in a unique and simple strategy easy to follow. Therefore, the main benefits of the proposed three-level ontology are specified as follows:

- The model specifies vocabulary in three levels of detail, which facilitates its reusability. Hence, an ontology designer is able to define and structure further vocabulary in the proper level of abstraction. Such capability is validated by means of extending the proposed vocabulary of each level of the model with the vocabulary taken from different scenarios. Such validation allowed us to identify that most of the scenarios were represented in a basic level of detail where several classes (entities or context information) were not depicted in the hierarchical structure needed to understand the semantic, definition and conceptualization of each of them. In this regard, the proposed model goes beyond such fact since the context knowledge is represented with the appropriate semantic given in the three levels of abstraction that allows identifying and describing the class, object, relation or individual that plays an important role in a real life scenario.
- Each level of the model and module of the middle-level ontology can be fully functional, i.e., they can be reused independently of the entire model. Hence, an ontology designer can integrate in an existing ontology the level of abstraction that is required without the dependency on other levels of the model (e.g. the upper-level ontology can be easily reused in other models to provide the semantic and structure of classes with other level of abstraction without any dependency). From the same perspective, an ontology designer can reuse any module of the middle-level ontology without inconsistencies since the semantic of each module is independent of each other (e.g., the module of time can be reused without the integration of other modules since it was formulated to represent only the semantic of time). Such capability is validated by evaluating inconsistencies of each module and level of the proposed model with the engine of Protégé.
- The lower-level ontology and modules of the middle-level ontology can be integrated with other modules or domain-dependent ontologies maintaining the semantic reasoning defined in upper levels. For this purpose, the ontology designer should follow a formal integration methodology as the presented in the development of the middle-level ontology to avoid inconsistencies in the generated axioms. Such capability is validated through the integration process of modules and levels of the ontology.

From the usability point of view, the model maintains a small size providing basic ontological resources avoiding integrate irrelevant vocabulary especially identified in large foundational ontologies that have been reused for purposes of context modelling. However, although the model aims at consolidating basic and relevant context knowledge following different criteria and aspects, it can be extended and reused in different directions. First, due to the intuitively visual representation and the detailed considerations in the integration process the upper and middle levels can be easy extended or modified (e.g. the user can decide to maintain or interchange a module). Second, each level and modules of the model has been implemented separately into the Prótegé editor in OWL to facilitate the reuse of the proposed resources. The implemented resources are available at: https://github.com/ocabgit/Three-LevelContextOntology.git.

We believe that the proposed three-level ontology can be used as a reference context model, in order to alleviate different issues such as the provisioning and establishment of replicated context hierarchies, inconsistencies on context knowledge definitions, among others. The context model is intended to be adequately generic encompassing all the present and future particularities of context knowledge.

RQ3) How to acquire? To acquire and provide reliable context data that can be later processed, a monitoring infrastructure with the desired functionality and quality levels is needed. For this purpose, this infrastructure should support the constant changes in its context and the context of other entities, including the services that it is supervising. To cope this challenge, we proposed CAMA, a context-aware monitoring architecture for context acquisition and dissemination that can be easily configured, adapted or evolved according to these changes.
CAMA has been designed as a service-oriented architecture and builds upon the proposed context ontology for modelling inputs, outputs and capabilities of monitoring tools, and data management. The proposed context-aware monitoring architecture has been developed for supporting highly dynamic environments that affect different entities including the monitoring infrastructure, services and applications. For this purpose, CAMA answers the following sub-RQs as follows:

RQ3.1 How can we provide a context-aware monitoring tool with the required reconfiguration and adaptation capabilities to face the constantly changing situation of the services and applications that compose a context-aware system? The architecture provides the capabilities of replacing, adding, reconfiguring and adapting monitoring tools by proving a decoupled architecture based on services that can be orchestrated in collaboration with the capabilities offered by an ontology including reasoning and the provisioning of a clear schema of parameters that can be configured for each monitoring tool integrated in the architecture. CAMA was developed following SOA principles that allow extending each component of the architecture with different methods and techniques independently of the approach used to obtain, process and store the data. Hence, the users can add through a common interface, different monitoring tools for monitoring a portfolio of channels (e.g., social networks, marketplaces, etc.).

RQ3.2 How can we structure, store and manage the data gathered by a context-aware monitoring tool in order to be easily disseminated to other components? The architecture provides the capability to access the monitored data using a rich semantic method since it was designed for collaborating with a context ontology in order to retrieve and provide a unified and representative schema of the monitored data produced by the monitoring tools and its related configuration instances.

We have also presented a state of the art on monitoring approaches revealing that researchers and practitioners have developed several tools and methods to support monitoring. However, as the study indicates, there is a lack of combined approaches considering different approaches of monitoring information in software projects. To cope this challenge, CAMA also provides a component named "group of monitors" that can be useful for plugging and playing different strategies of monitoring. Such component is supported by a quality model for analyzing and selecting the most appropriate monitoring tools given certain application domain. The model was developed based on the characterization of different types of monitoring approaches identified through a state of the art of monitoring tools. At this respect, a set of commercial monitoring tools identified in such state of the art were selected given the features that they presented and evaluated in the proposed quality model demonstrating its usability.

Finally, we have integrated the results of RQ2 and RQ3 in a context-ware framework with the purpose of supporting the whole context life cycle: acquisition, modelling, reasoning, and distribution. In this regard, the context ontology takes an active role for reconfiguring and adapting the context-aware monitoring architecture (CAMA).

For demonstrating the feasibility and potential of the results derived from RQ2 and RQ3 of the present PhD thesis different validations were conducted, described as follows:

RQ2. Three-level context ontology (3LConOnt). A) We modelled an abstract layer ontology for conceptualizing any monitoring tool based on the resources provided by the 3LConOnt, and as an example of its usage, we built a domain ontology for characterizing social networks monitoring tools. B)

We evaluated the level of generality of the 3LConOnt by validating its reusability, extensibility and adaptation through different smart scenarios. C) We evaluated its consistency and reasoning capabilities by triggering queries to the proposed model based on some competence questions and from the perspective of a smart restaurant scenario. D) We evaluated its reusability in existing context ontologies demonstrating that each level or module of the model is independent of the entire model. E) We evaluated its correctness and completeness for demonstrating that the ontology is consistent and copes the vocabulary needed in the modules. The results obtained through these validations have successful demonstrated the main benefits of the proposed model, fulfilling different capabilities of a consistent model such as generality, reusability, integration, extensibility, completeness, etc.

RQ3. Context-aware monitoring architecture (CAMA). A) We assessed that CAMA provides the capability to add new monitors with minimal effort. In this case, we validated the feasibility of the proposed abstract layer ontology for conceptualizing monitoring tools in the core of CAMA by means of an empirical study where different criteria were evaluated allowing to conclude that the time consuming by different participants for modelling monitoring tools taking as basis the proposed abstract layer ontology is really low. Furthermore, the level of expertise in both modelling ontologies and monitoring systems was not an important variable to manage the proposed ontology and supporting material, it means that anyone can immediately start using the ontology without having to deeply and carefully studying it. B) We assessed that CAMA provides the capability to reconfigure and to replace monitors automatically. In this case, we implemented different monitoring tools and executed some validation protocols that demonstrated successful reconfigurations and replacements of the monitors. C) We assessed that CAMA provides the capability to access the monitored data using a rich semantics method following an open-world assumption. In this case, we executed queries and assertions through reasoning that have proven successful the consistency of the ontology and its reasoning capabilities.

RQ2 and RQ3. Context-aware framework. We designed a prototype that implements the framework to validate reconfigurations of context-aware monitors. In particular, we implemented a self-reconfigurable monitor of twitter, and run it during the final tie-breaker games of the chess championship.

In general, the validations specified above were aligned to the requirements of a European H2020 project named SUPERSEDE. Specifically, based on the use cases of the project namely monitoring social networks and monitoring IT infrastructure. Such scenarios were focused on delivering monitoring methods and tools to support decision-making in the evolution and adaptation of software services and applications ultimately leading to improved end-users' quality of experience (QoE).

As future work, we plan:

1) To validate the resulting modules and levels of the proposed three-level ontology that we got by hand by comparing them with modules and levels obtained in tools that automatically retrieve parts of an ontology.
2) To map the context knowledge pieces of the proposed context model with a well-structured and well-defined concepts of a foundational ontology in order to reduce semantic inconsistencies between them.
3) To evaluate the context model in different actual and trending cases of smart cities and internet of things to unify and increase the semantic and meaning of the data obtained from sensors and exchanged among different real context-aware services.
4) To evaluate the performance overhead of the ontology in practice.
5) To address the integration and collaboration of different types of logical, physical and logical monitors for evolving the components and interfaces provided in the monitoring architecture.
6) To develop a management system that can operate different tasks over the proposed monitoring architecture, i.e., assigning and reconfiguring monitors as well as providing dashboards of the monitored data.

7) To apply the resulting context model and monitoring architecture in different use cases of the context-awareness research area. For this purpose, we are planning to design different context-aware scenarios including ranking and adaptation of services and applications; evolution and adaptation of personalized software; improving the QoE of the user; etc.
8) To obtain context information from different information systems that obtain raw data from different sensors with the aim of stressing the capabilities of the context model and the monitoring architecture, and therefore identify both, lacks and capabilities for identifying, creating and disseminating complex context information.
9) To develop different smart apps based on the capabilities of the context model and the monitoring architecture to consolidate the usability of both. For instance, a smart app can retrieve the coordinates of a location and the reasoning module can deduce some aspects of the environment.
10) To enhance the context-aware framework in several directions. On the one hand, we plan to implement more context-aware monitors to demonstrate the approach in more complex scenarios. On the other hand, we plan to improve the usability of the framework by using a higher-level language to specify conditions in order to abstract the user from low-level technical aspects of SWRL.

"If one cannot enjoy reading a book over and over again, there is no use in reading it at all"

- Oscar Wilde

CHAPTER 7

Bibliography

[1] Bazire, M., Brézillon, Patrick.: Understanding Context Before Using It. *In: 5th International Conference on Modeling and Using Context*, pp. 29-40 (2005)

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

[3] Dey, A.K.: Understanding and Using Context. *Journal of Personal and ubiquitous computing*. 5, 4-7 (2001)

[4] Korpipää, P., Jani, M., Kela, J., Malm, E.J.: Managing Context Information in Mobile Devices. *Journal of IEEE pervasive computing*. 3, 42-51 (2003)

[5] Hynes, G., Reynolds, V., Hauswirth, M.: A Context Lifecycle for Web-Based Context Management Services. *In: 4th European Conference on Smart Sensing and Context*, pp. 51-65 (2009)

[6] Gu, T., Pung, H.K., Zhang, D.Q.: A Service-Oriented Middleware for Building Context-Aware Services. *Journal of Network and Computer Applications*. 28, 1-8 (2005)

[7] Hong, J.Y., Suh, E.H., Kim, S.J.: Context-Aware Systems: A Literature Review and Classification. *Journal of Expert Systems with Applications*. 36, 8509-8522 (2009)

[8] Byun, H.E., Cheverst, K.: Utilizing Context History to Provide Dynamic Adaptations. *Journal of Applied Artificial Intelligence*. 18, 533–548 (2004)

[9] Dey, A.K.: Providing Architectural Support for Building Context-Aware Applications. *Ph.D. Dissertation*, Georgia (2000)

[10] Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context Aware Computing for the Internet of Things: A Survey. *Journal of Communications Surveys & Tutorials*. 16, 414-454 (2014)

[11] Henricksen, K.: A Framework for Context-Aware Pervasive Computing Applications. *Ph.D. Dissertation*, Queensland (2003)

[12] Cao, Y., Klamma, R., Hou, M., Jarke, M.: Follow Me, Follow You-Spatiotemporal Community Context Modeling and Adaptation for Mobile Information Systems. *In: 9th IEEE International Conference on Mobile Data Management*, pp. 108-115 (2008)

[13] Crowley, J.L., Coutaz, J., Rey, G., Reignier, P.: Perceptual Components for Context Aware Computing. *In: 4th ACM International Conference on Pervasive and Ubiquitous Computing*, pp. 117-134 (2002)

[14] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A Survey of Context Modelling and Reasoning Techniques. *Journal of Pervasive and Mobile Computing*. 6, 161-80 (2010)

[15] Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. *In: 1st International Workshop on Advanced Context Modelling, Reasoning and Management* (2004)

[16] Sudhana, K.M., Raj, V.C., Suresh, R.M.: An Ontology-Based Framework for Context-Aware Adaptive e-Learning System. *In: IEEE International Conference on Computer Communication and Informatics*, pp. 1-6 (2013)
[17] Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. *In: 1st IEEE International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 258-267 (2004)
[18] Noy, N.F.: Semantic Integration: A Survey of Ontology-Based Approaches. *Journal of ACM Sigmod Record*. 65-70 (2004)
[19] Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning Using OWL. *In: 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp. 18-22 (2004)
[20] Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. *Journal of Computer*. 40, 38-45 (2007)
[21] Bon, J.V.: Best Practices: Introduction to ITIL. *TSO Publications*, Norwich (2007)
[22] Badidi, E., Taleb, I.: Towards a Cloud-Based Framework for Context Management. *In: IEEE International Conference on Innovations in Information Technology*, pp. 35-40 (2011)
[23] Schmidt, A.: Context-Aware Computing: Context-Awareness, Context-Aware User Interfaces, and Implicit Interaction. *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. (2013)
[24] Shaw, M.: The Coming-of-Age of Software Architecture Research. *In: 23rd IEEE International Conference on Software Engineering*, pp. 656 (2001)
[25] Cabrera, O., Franch, X., Marco, J.: Ontology-Based Context Modelling in Service-Oriented Computing: A Systematic Mapping. *Journal of Data & Knowledge Engineering* (2017). To appear
[26] Cabrera, O., Franch, X., Marco, J.: A Context Ontology for Service Provisioning and Consumption. *In: 8th IEEE International Conference on Research Challenges in Information Science (RCIS)*, pp. 1-12 (2014)
[27] Cabrera, O., Franch, X., Marco, J.: A Middle-Level Ontology for Context Modelling. *In: 34th International Conference on Conceptual Modeling*, pp. 148-156 (2015)
[28] Nadal, S., Romero, O., Abelló, A., Cabrera, O.: D2.2: Data Management. SUPERSEDE deliverable, (2016)
[29] Cabrera, O., Franch, X., Marco, J.: 3LConOnt: A Three-Level Ontology for Context Modelling in Context-Aware Computing. *Journal of Software and Systems Modeling*. (2016). To appear
[30] Cabrera, O., Guzmán, E., Marco, J., Perini, A., Seyff, N., Stade, M.: D1.1: Feedback Management and Monitoring Approaches. SUPERSEDE deliverable, (2015)
[31] Oriol, Marc., Marco, J., Cabrera, O., Motger, Q., Fotrousi, F., Muñante, D., Perini, A.: D1.4: Comprehensive Monitoring Techniques. SUPERSEDE deliverable, (2016)
[32] Cabrera, O., Franch, X.: A Quality Model for Analyzing Web Service Monitoring Tools. *In: 6th IEEE International Conference on Research Challenges in Information Science (RCIS)*, pp. 1-12 (2012)
[33] Cabrera, O., Oriol, M., Franch, X., Marco, J.: CAMA: A Context-Aware Monitoring Architecture for Highly Dynamic Environments. *Submitted to journal of Expert Systems with Applications*. (2016)
[34] Cabrera, O., Oriol, M., Franch, X., Marco, J., López, L., Fragoso, O., Santaolaya, R.: Open Framework for Web Service Selection Using Multimodal and Configurable Techniques. *Journal of Computación y Sistemas*. 18, 665-682 (2014)

[35] Cabrera, O., Oriol, M., Franch, X., Marco, J., López, L., Fragoso, O., Santaolaya, R.: WeSSQoS: A Configurable SOA System for Quality-Aware Web Service Selection. Technical Report, arXiv (2011)

[36] Cabrera, O., Oriol, M., Franch, X., Marco, J., López, L., Fragoso, O., Santaolaya, R.: WeSSQoS: Un Sistema SOA para la Selección de Servicios Web según su Calidad. *In: Jornadas Científico-Técnicas en Servicios Web y SOA*, pp. 76-89 (2009)

[37] Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update. *Journal of Information and Software Technology*. 64, 1-18 (2015).

[38] Kitchenham, B. Charters, S.: Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical report, EBSE (2007)

[39] Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O.P., Turner, M., Niazi, M., Linkman, S.: Systematic Literature Reviews in Software Engineering-A Tertiary Study. *Journal of Information and Software Technology*. 52, 792-805 (2010)

[40] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic Mapping Studies in Software Engineering. *In: 12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 1-10 (2008)

[41] Bellavista, P., Corradi, A., Fanelli, M., Foschini, L.: A Survey of Context Data Distribution for Mobile Ubiquitous Systems. *Journal of ACM Computing Surveys*. 44, 24 (2012)

[42] Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Technical report, TR2000-381. Dept. of Computer Science, Dartmouth College (2000)

[43] Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-Aware Systems. *Journal of Ad Hoc and Ubiquitous Computing*. 2, 263-277 (2007)

[44] Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F.A., Tanca, L.: A Data-Oriented Survey of Context Models. *Journal of ACM Sigmod Record*. 36, 19-26 (2007)

[45] Ye, J., Coyle, L., Dobson, S., Nixon, P.: Ontology-Based Models in Pervasive Computing Systems. *Journal of the Knowledge Engineering Review*. 22, 315-347 (2007)

[46] Kjaer, K.E.: A Survey of Context-Aware Middleware. *In: 25th Conference on IASTED International Multi-Conference: Software Engineering*, pp. 148-155 (2007)

[47] Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C., Nurcan, S.: Semantic Representation of Context Models: A Framework for Analyzing and Understanding. *In: 1st ACM Workshop on Context, Information and Ontologies*, pp. 6 (2009)

[48] Perttunen, M., Riekki, J., Lassila, O.: Context Representation and Reasoning in Pervasive Computing: Review. *Journal of Multimedia and Ubiquitous Engineering*. 4, 1-28 (2009)

[49] Saeed, A., Waheed, T.: An Extensive Survey of Context-Aware Middleware Architectures. *In: IEEE International Conference on Electro/Information Technology*, pp. 1-6 (2010)

[50] Vanathi, B., Uthariaraj, R.V.: Ontology Based Service Discovery for Context Aware Computing. *In: 1st IEEE International Conference on Advanced Computing*, pp. 51-54 (2009)

[51] Bauer, C.: A Comparison and Validation of 13 Context Meta-Models. *In: European Conference on Information Systems*, pp. 17 (2012)

[52] Makris, P., Skoutas, D.N., Skianis. C.: A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration. *Journal of IEEE Communications Surveys & Tutorials*. 15, 362-386 (2013)

[53] Schuster, D., Rosi, A., Mamei, M., Springer, T., Endler, M., Zambonelli, F.: Pervasive Social Context: Taxonomy and Survey. *Journal of ACM Transactions on Intelligent Systems and Technology*. 4, 46 (2013)

[54] Koç, H., Hennig, E., Jastram, S., Starke, C.: State of the Art in Context Modelling-A Systematic Literature Review. *In: Advanced Information Systems Engineering Workshops*, pp. 53-64 (2014)

[55] Kitchenham, B., Brereton, P., Turner, M., Niazi, M., Linkman, S., Pretorius, R., Budgen, D.: The Impact of Limited Search Procedures for Systematic Literature Reviews-A Participant-Observer

Case Study. *In: 3rd IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 336-345 (2009)
[56] Dieste, O., Grimán, A., Juristo, N.: Developing Search Strategies for Detecting Relevant Experiments. *Journal of Empirical Software Engineering*. 14, 513-539 (2009)
[57] Kitchenham, B., Mendes, E., Travassos, G.H.: Cross Versus Within-Company Cost Estimation Studies: A Systematic Review. *Journal of IEEE Transactions on Software Engineering*. 33, 316-329 (2007)
[58] Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: From Ontological Art Towards Ontological Engineering. *In: Spring Symposium on Ontological Engineering of AAAI*, pp. 33–40 (1997)
[59] Cabrera, O., Franch, F., Marco, J.: Online Annexes of Context Modeling in Service-Oriented Computing: A Systematic Mapping. http://gessi.lsi.upc.edu/qcontextmodels/
[60] Maslow, A.H., Frager, R., Cox, R.: Motivation and Personality. Harper & Row, New York (1970)
[61] Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. *Journal of the Knowledge Engineering Review*. 13, 31-89 (1998)
[62] Lenat, D.B., Guha, R.V.: Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project. Addison Wesley Longman Publishing Co., Inc., Boston (1989)
[63] Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Laerhoven, K.V., Velde W.V.: Advanced Interaction in Context. In: 1st International Symposium on Handheld and ubiquitous computing, pp. 89-101 (1999)
[64] Schmidt, A., Beigl, M., Gellersen, H.W.: There is More to Context than Location. *Journal of Computers & Graphics*. 23, 893-901 (1999)
[65] Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. *In: 1st IEEE Workshop on Mobile Computing Systems and Applications*, pp. 85-90 (1994)
[66] Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling Context Information in Pervasive Computing systems. *In: 1st International Conference on Pervasive Computing*, pp. 167-180 (2002)
[67] Held, A., Buchholz, S., Schill, A.: Modeling of Context Information for Pervasive Computing Applications. *In: 6th World Multiconference on Systemics, Cybernetics and Informatics*, pp. 6 (2002)
[68] Tarasewich, P.: Towards a Comprehensive Model of Context for Mobile and Wireless Computing. *In: 9th Americas Conference on Information Systems*, pp. 114-124 (2003)
[69] Ghannem, A., Hamdi, M., Abdelmoez, W., Ammar, H.: A Context Model Development Process for Smart City Operations. *In: IEEE International Conference on Service Operations and Logistics, and Informatics*, pp. 122-127 (2015)
[70] Mok, J., Min, H.: Ontology-Based Context-Aware Model by Applying Bayesian Network. *In: 12th IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2660-2664 (2015)
[71] Devaraju, A., Hoh, S.: Ontology-Based Context Modeling for User-Centered Context-Aware Services Platform. *In: IEEE International Symposium on Information Technology*, pp. 1-7 (2008)
[72] Pietschmann, S., Mitschick, A., Winkler, R., Meißner. K.: Croco: Ontology-Based, Cross-Application Context Management. *In: 3rd IEEE International Workshop on Semantic Media Adaptation and Personalization*, pp. 88-93 (2008)
[73] Cadenas, A., Ruiz, C., Larizgoitia, I., García-Castro, R., Lamsfus, C., Vázquez, I., González, M., Martín, D., Poveda, M.: Context Management in Mobile Environments: A Semantic Approach. *In: 1st ACM Workshop on Context, Information and Ontologies*, pp. 2 (2009)
[74] Rubio, G., Serral, E., Castillejo, P., Martínez, J.F.: A Novel Context Ontology to Facilitate Interoperation of Semantic Services in Environments with Wearable Devices. *In: Workshops On the Move to Meaningful Internet Systems*, pp. 495-504 (2012)
[75] Hobbs, J.R., Pan, F.: An Ontology of Time for the Semantic Web. *In: ACM Transactions on Asian Language Information Processing*, pp. 66-85 (2004)
[76] Niles, I., Pease, A.: Towards a Standard Upper Ontology. *In: International Conference on Formal Ontology in Information Systems*, pp. 2-9 (2001)

[77] Ankolenkar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D.L., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T.R., Sycara, K.: OWL-S: Semantic Markup for Web Services (Version 1.0). Specification, OWL Services Coalition (2003)

[78] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., Retschitzegger, W.: Context-Awareness on Mobile Devices-the Hydrogen Approach. *In: 36th IEEE Annual Hawaii International Conference on System Sciences*, pp. 10 (2003)

[79] Preuveneers, D., den Bergh, J.V., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., de Bosschere, K.: Towards an Extensible Context Ontology for Ambient Intelligence. *In: 2nd European Symposium on Ambient intelligence*, pp. 148-159 (2004)

[80] Heckmann, D.: Situation Modeling and Smart Context Retrieval with Semantic Web Technology and Conflict Resolution. *In: 2nd International Workshop on Modeling and Retrieval of Context*, pp. 34-47 (2006)

[81] Chen, H., Finin, T., Joshi, A.: An Ontology for Context-Aware Pervasive Computing Environments. *Journal of the Knowledge Engineering Review*. 18, 197-207 (2003)

[82] Xynogalas, S., Roussaki, I., Chantzara, M., Anagnostou, M.: Context Management in Virtual Home Environment Systems. *Journal of Circuits, Systems, and Computers*. 13, 293-311 (2004)

[83] Bu, Y., Li, J., Chen, S., Tao, X., Lv, J.: An enhanced ontology based context model and fusion mechanism. *In: International Conference on Embedded and Ubiquitous Computing*, pp. 920-929 (2005)

[84] Kim, E., Choi, J.: An Ontology-Based Context Model in a Smart Home. *In: International Conference on Computational Science and Its Applications*, pp. 11-20 (2006)

[85] Neto, R.F.B., Pimentel, M.G.C.: Toward a Domain-Independent Semantic Model for Context-Aware Computing. *In: 3rd IEEE Latin American Web Congress*, pp. 10 (2005)

[86] Park, T.H., Kwon, O.: Identifying a Generic Model of Context for Context-Aware Multi-Services. *In: 4th International Conference on Ubiquitous Intelligence and Computing*, pp. 919-928 (2007)

[87] Brickley, D., Miller, L.: FOAF vocabulary specification. Namespace Document 9 (2003)

[88] Ngo, Q., Shehzad, A., Liaquat, S., Riaz, M., Lee, S.: Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework. *In: International Conference on Embedded and Ubiquitous Computing*, pp. 672-681 (2004)

[89] Bradley, N.A., Dunlop, M.D.: Toward a Multidisciplinary Model of Context to Support Context-Aware Computing. *Journal of Human-Computer Interaction*. 20, 403-446 (2005)

[90] Kritsotakis, M., Michou, M., Nikoloudakis, E., Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D.: C-ngine: A Contextual Navigation Guide for Indoor Environments. *In European Conference on Ambient Intelligence*, pp. 258-275 (2008)

[91] Sheng, J., Zou, W., Yang, L., Wang, B.: A RFID-Based Context-Aware Service Model. In: 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 779-784 (2011)

[92] Corcho, O., Fernández-López, M., Gómez-Pérez, A., López-Cima, A.: Building Legal Ontologies with METHONTOLOGY and WebODE. *In: Law and the Semantic Web*, pp. 142-157 (2005)

[93] Lee, H., Kwon, J.: Ontology Model-Based Situation and Socially-Aware Health Care Service in a Smart Home Environment. *Journal of Smart Home*. 7, 239-250 (2013)

[94] Chaker, H., Chevalier, M., Soulé-Dupuy, C., Tricot, A.: Improving Information Retrieval by Modeling Business Context. *In: 3rd IEEE International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services*, pp. 117-122 (2010)

[95] Uschold, M., Gruninger, M.: Ontologies: Principles, Methods and Applications. *Journal of Knowledge Engineering Review*. 11, 93-136 (1996)

[96] Hervás, R., Bravo, J., Fontecha, J.: A Context Model Based on Ontological Languages: A Proposal for Information Visualization. *Journal of Universal Computer Science*. 16, 1539-1555 (2010)

[97] Prekop, P., Burnett, M.: Activities, Context and Ubiquitous Computing. *Journal of Computer Communications*. 26, 1168-1176 (2003)

[98] Khedr, M., Karmouch, A.: ACAI: Agent-Based Context-Aware Infrastructure for Spontaneous Applications. *Journal of Network and Computer Applications*. 28, 19-44 (2005)

[99] Gong, R., Ning, K., Li, Q., O'Sullivan, D., Chen, Y., Decker, S.: Context Modeling and Measuring for Proactive Resource Recommendation in Business Collaboration. *Journal of Computers & industrial engineering*. 57, 27-36 (2009)

[100] Villegas, N.M., Müller, H.A.: The Smartercontext Ontology and Its Application to the Smart Internet: A Smarter Commerce Case Study. *In: Chignell, M., Cordy, J.R., Kealey, R., Ng, J., Yesha, Y. (eds.) The Personal Web. Lecture Notes in Computer Science, vol. 7855*. Springer 151-184 (2013)

[101] Strassner, J., Liu, Y., Jiang, M., Zhang, J., Meer, S.V., Foghlú, M., Fahy, C., Willie Donnelly, W.: Modelling Context for Autonomic Networking. *In: IEEE Network Operations and Management Symposium Workshops*, pp. 299-308 (2008)

[102] Zimmermann, A., Lorenz, A., Oppermann, R.: An Operational Definition of Context. *In: 6th International and Interdisciplinary Conference on Modeling and Using Context*, pp. 558-571 (2007)

[103] Shen, H., Cheng, Y.: A Semantic Context-Based Model for Mobile Web Services Access Control. *Journal of Computer Network and Information Security*. 1, 18-25 (2011)

[104] Kayes, A.S.M, Han, J., Colman, A.: An Ontology-Based Approach to Context-Aware Access Control for Software Services. *In: 14th International Conference on Web Information Systems Engineering*, pp. 410-420 (2013)

[105] Śliwa, J., Gleba, K.: Semantic Description of the Context of the Service Call. *In: Park J., Stojmenovic I., Jeong H., Yi G. (eds.) Computer Science and its Applications. Lecture Notes in Electrical Engineering, vol 330.* Springer, 483-489 (2015)

[106] Krömker, H., Wienken, T.: Context Elicitation for User-Centered Context-Aware Systems in Public Transport. *In: International Conference on Human-Computer Interaction*, pp. 429-439 (2015)

[107] Aly, M., Elbarachi, M., Dssouli, R.: A New Information Model Towards Context-Aware Service Provisioning in the Internet-of-Things. *In: 18th IEEE International Conference on Intelligence in Next Generation Networks*, pp. 160-167 (2015)

[108] Khemaja, M., Buendía, F.: An Ontology-Driven Approach to Model & Support Mobility and GeoLocation Based Campus Interactions. *In: 16th ACM International Conference on Human Computer Interaction*, pp. 9 (2015)

[109] Hoque, M., Kabir, M., Thapa, K., Yang, S.: Ontology-based Context Modeling to Facilitate Reasoning in a Context-Aware System: A Case Study for the Smart Home. *Journal of Smart Home*. 9, 151-156 (2015)

[110] HameurLaine, A., Abdelaziz, K., Roose, P., Kholladi, M.: Ontology and Rules-Based Model to Reason on Useful Contextual Information for Providing Appropriate Services in U-Healthcare Systems. *In: Camacho D., Braubach L., Venticinque S., Badica C. (eds.) Intelligent Distributed Computing VIII. Studies in Computational Intelligence, vol 570.* Springer, 301-310 (2015)

[111] Khouja, M., Juiz, C.: Enhanced Service Discovery Via Shared Context in a Distributed Architecture. *In: IEEE International Conference on Web Services*, pp. 273-280 (2015)

[112] Liang-Liang, C.: Context-Aware Technology Research in WoT Business Environment. *In: 7th IEEE International Conference on Measuring Technology and Mechatronics Automation*, pp. 310-313 (2015)

[113] Saidani, O., Rolland, C., Nurcan, S.: Towards a Generic Context Model for BPM. *In: 48th IEEE Hawaii International Conference on System Sciences*, pp. 4120-4129 (2015)

[114] Restrepo, S.E., Pezoa, J.E., Carranza, D.O.: An Adaptive Architecture for Ambient Intelligence Based on Meta-Modeling, Smart Agents, and Wireless Sensor Networks. *Journal of IEEE Latin America Transactions*. 12, 1508-1514 (2014)

[115] Lee, M., Bak, C., Lee, J.W.: A Prediction and Auto-Execution System of Smartphone Application Services Based on User Context-Awareness. *Journal of Systems Architecture*. 60, 702-710 (2014)

[116] Xue, X., Chang, J., Liu, Z.: Context-Aware Intelligent Service System for Coal Mine Industry. *Journal of Computers in Industry*. 65, 291-305 (2014)
[117] Wang, B., Tang, X.: Designing a Self-Adaptive and Context-Aware Service Composition System. *In: IEEE Conference on Computing, Communications and IT Applications*, pp. 155-160 (2014)
[118] Sohn, M., Jeong, S., Lee, H.J.: Case-Based Context Ontology Construction Using Fuzzy Set Theory for Personalized Service in a Smart Home Environment. *Journal of Soft Computing*. 18, 1715-1728 (2014)
[119] Yus, R., Pappachan, P., Das, P.K., Finin, T., Joshi, A., Mena, E.: Semantics for Privacy and Shared Context. *In: CEUR Proceedings of the Workshop on Society, Privacy and the Semantic Web-Policy and Technology*, pp. 34-35 (2014)
[120] Choi, J., Kim, J., Kim, P.: Modeling for Context-Aware Healthcare Service Using Ontology. *In: 29th Annual ACM Symposium on Applied Computing*, pp. 1729-1730 (2014)
[121] Sotsenko, A., Jansen, M., Milrad, M.: Using a Rich Context Model for a News Recommender System for Mobile Users. *In: 2nd International Workshop on News Recommendation and Analytics in conjunction with 22nd Conference on User Modelling, Adaptation and Personalization*, (2014)
[122] Li, S., Chen, H.: A Context-Aware Framework for SaaS Service Dynamic Discovery in Clouds. *In: 14th International Conference on Algorithms and Architectures for Parallel Processing*, pp. 671-684 (2014)
[123] Chellouche, S.A., Négru, D., Arnaud, J., Batalla, J.M.: Context-Aware Multimedia Services Provisioning in Future Internet Using Ontology and Rules. *In: International Conference and Workshop on the Network of the Future*, pp. 1-5 (2014)
[124] Zhang, Y., Wang, J., Yan, Y.: Context-Aware Generic Service Discovery and Service Composition. *In: IEEE International Conference on Mobile Services*, pp. 132-139 (2014)
[125] Berri, J.: Context Reasoning for Mobile Systems. *In: IEEE World Congress on Computer Applications and Information Systems*, pp. 1-5 (2014)
[126] Li, P.S., Liu, A., Zhou, P.C.: Context Reasoning for Smart Homes Using Case-Based Reasoning. *In: 18th IEEE International Symposium on Consumer Electronics*, pp. 1-2 (2014)
[127] Katsumata, M.: Task Context-Aware e-Mail Platform for Collaborative Tasks. *Journal of Human-Centric Computing and Information Sciences*. 4, 1-10 (2014)
[128] Da, K., Roose, P., Dalmau, M., Nevado, J., Karchoud, R.: Kali2Much: A Context Middleware for Autonomic Adaptation-Driven Platform. *In: 1st ACM Workshop on Middleware for Context-Aware Applications in the IoT*, pp. 25-30 (2014)
[129] Benlamri, R., Zhang, X.: Context-Aware Recommender for Mobile Learners. *Journal of Human-centric Computing and Information Sciences*. 4, 12 (2014)
[130] Miraoui, M.: Dynamic Adaptation in Ubiquitous Services: A Conceptual Architecture. *In: Handbook of Research on Architectural Trends in Service-Driven Computing*, pp. 160-180 (2014)
[131] Rahman, M., Hossain, M., Saddik, A.: Context-Aware Multimedia Services Modeling: An e-Health Perspective. *Journal of Multimedia Tools and Applications*. 73, 1147-1176 (2014)
[132] Diallo, B., Badard, T., Hubert, F., Daniel, S.: Context-Based Mobile GeoBI: Enhancing Business Analysis with Contextual Metrics/Statistics and Context-Based Reasoning. *Journal of GeoInformatica*. 18, 405-433 (2014)
[133] Maalej, M., Mtibaa, A., Gargouri, F.: Ontology-Based User Modeling for Handicraft Woman Recommendation. *In: International Conference on Model and Data Engineering*, pp. 138-145 (2014)
[134] Furno, A., Zimeo, E.: Context-Aware Composition of Semantic Web Services. *Journal of Mobile Networks and Applications*. 19, 235-248 (2014)
[135] Novakovic, D., Huemer, C.: Putting Services in Context (Short Paper). *In: 6th IEEE International Conference on Service-Oriented Computing and Applications*, pp. 38-42 (2013)

[136] Ai, D.X., Zuo, H., Yang, J.: Personalized Mobile Catering Recommender System Based on Context Ontology Model and Rule Inference. *Journal of Advanced Materials Research*. 717, 708-713 (2013)

[137] Boudaa, B., Hammoudi, S., Chikh, M.A.: ODM-Based Modeling for User-Centered Context-Aware Mobile Applications. *In: 3rd IEEE International Conference on Information Technology and e-Services*, pp. 1-7 (2013)

[138] Kim, H., Jeong, S., Kim, H., Yoe, H.: A Study on the Context Service Model for Livestock Estrus Detection. *Journal of Smart Home*. 7, 177-184 (2013)

[139] Fissaa, T., Guermah, H., Hafiddi, H., Nassar, M., Kriouile, A.: Ontology Based Context Modeler for Context-Aware Systems (Short paper). *In: 6th IEEE International Conference on Service-Oriented Computing and Applications*, pp. 43-47 (2013)

[140] Zouhaier, L., Hlaoui, Y., Ayed, L.: Building Adaptive Accessible Context-Aware for User Interface Tailored to Disable Users. In: 37th IEEE Annual Computer Software and Applications Conference Workshops, pp. 157-162 (2013)

[141] Kabir, M.A., Han, J., Yu, J., Colman, A.: SCIMS: A Social Context Information Management System for Socially-Aware Applications. *In: 24th International Conference on Advanced Information Systems Engineering*, pp. 301-317 (2012)

[142] Paul-Stueve T., Wachsmuth, S.: Towards a Social Context Model and Architecture for Large-Scale Pervasive Environments. *In: IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 619-624 (2012)

[143] Hu, B., Chibani, A., Amirat, Y.: Semantic Context Relevance Assessment in Urban Ubiquitous Environments. *In: ACM Conference on Ubiquitous Computing*, 639-640 (2012)

[144] Copeland, R., Crespi, N.: Implementing an Enterprise Business Context Model for Defining Mobile Broadband Policy. *In: 8th International Conference on Network and Service Management, International Federation for Information Processing*, pp. 209-213 (2012)

[145] Wong, S., Liu, D.: A Context Information Management Model for Tour Mobile E-commerce. *In: He X., Hua E., Lin Y., Liu X. (eds.) Computer, Informatics, Cybernetics and Applications. Lecture Notes in Electrical Engineering, vol 107*. Springer, 1443-1452 (2012)

[146] Jianfeng, X., Dong, W.: Object-Oriented and Ontology Context-Aware Modeling Based on XML. *In: 2nd IEEE International Conference on Computer Science and Network Technology*, pp. 1795-1800 (2012)

[147] Madkour, M., Maach, A.: Ontology-Based Context Modeling for Vehicle Context-Aware Services. *Journal of Theoretical and Applied Information Technology*. 34, 158-166 (2011)

[148] Jang, C., Choi, E.: Context Model Based on Ontology in Mobile Cloud Computing. *In: 3rd International Conference on Advanced Communication and Networking*, pp. 146-151 (2011)

[149] Gilman, E., Su, X., Riekki, J.: Towards Context Modeling and Reasoning in a Ubiquitous Campus. *In: Conference on Information Modelling and Knowledge Bases*, pp. 278-287 (2011)

[150] Wusheng, W., Weiping, L., Zhonghai, W., Weijie, C., Tong, M.: An Ontology-Based Context Model for Building Context-Aware Services. *In: 2nd IEEE International Conference on Intelligent Systems, Modelling and Simulation*, pp. 296-299 (2011)

[151] Li, M.: Ontology-Based Context Information Modeling for Smart Space. *In: 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing*, pp. 278-283 (2011)

[152] Kim, L., Kim, H., Yoon, Y.: Fusion Context Model Based on User Scenario for Smart Service. *In: International Conference on Grid and Distributed Computing*, pp. 506-514 (2011)

[153] Changboka, Chang, H., Ahn, H., Choi, E.: Efficient Context Modeling Using OWL in Mobile Cloud Computing. *Journal of Energy Procedia*. 16, 1312-1317 (2011)

[154] Durán, J.I., Laitakari, J., Pakkala, D., Perälä, J.: A User Meta-Model for Context-Aware Recommender Systems. In: 1st ACM International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 63-66 (2010)

[155] Tong, M.W., Liu, Q.T., Liu, X.N.: A Service Context Model Based on Ontology for Content Adaptation in e-Learning. *In: IEEE Frontiers in Education Conference*, pp. S1D-1 (2010)

[156] Luo, S., Xu, B., Sun, K.: Compose Real Web Services with Context. *In: IEEE International Conference on Web Services*, pp. 630-631 (2010)
[157] Yi, Z., Fan, Z., Mingfeng, C., Chao, D.: Notice of Retraction Context Aware-Based Learning Resources Retrieval Model for Ubiquitous Learning Environment. *In: 2nd IEEE International Conference on Future Computer and Communication*, V1-590 (2010)
[158] Sun, J., Zhang, Y., He, K.: A Petri-Net Based Context Representation in Smart Car Environment. *In: 5th International Conference on Advances in Grid and Pervasive Computing*, pp. 162-173 (2010)
[159] Sigg, S., Haseloff, S., David, K.: An Alignment Approach for Context Prediction Tasks in UbiComp Environments. *Journal of IEEE Pervasive Computing*. 9, 90-97 (2010)
[160] Park, Y., Moon, A., Choi, Y., Kim, S., Kim, S.: An Efficient Context Model for Fast Responsiveness of Context-Aware Services in Mobile Networks. *In: 7th IEEE Consumer Communications and Networking Conference*, pp. 1-5 (2010)
[161] Zainol, Z., Nakata, K.: Generic Context Ontology Modeling: A Review and Framework. *In: 2nd IEEE International Conference on Computer Technology and Development*, 126-130 (2010)
[162] Saeedi, S., El-Sheimy, N., Malek, M., Samani, N.: An Ontology Based Context Modeling Approach for Mobile Touring and Navigation System. *In: Canadian Geomatics Conference and Symposium of Commission I*, pp. 15-18 (2010)
[163] Bandara, K.Y., Wang, M., Pahl, C.: Context Modeling and Constraints Binding in Web Service Business Processes. *In: 1st ACM International Workshop on Context-Aware Software Technology and Applications*, pp. 29-32 (2009)
[164] Liu, M., Corno, F., Bonino, D., Castellina, E.: An Ontology-Based Context Management and Reasoning on the DOG Gateway. *In: IEEE International Conference on Computational Intelligence and Software Engineering*, pp. 1-4 (2009)
[165] Hu, Y., Li, X.: An Ontology Based Context-Aware Model for Semantic Web Services. *In: 2nd IEEE International Symposium on Knowledge Acquisition and Modeling*, 426-429 (2009)
[166] Cioara, T., Anghel, I., Salomie, I., Dinsoreanu, M.: A Context-Based Semantically Enhanced Information Retrieval Model. *In: 5th IEEE International Conference on Intelligent Computer Communication and Processing*, pp. 245-250 (2009)
[167] Cao, Y., Tao, L., Xu, G.: An Event-Driven Context Model in Elderly Health Monitoring. *In: IEEE Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pp. 120-124 (2009)
[168] Lamsfus, C., Alzua-Sorzabal, A., Martin, D., Salvador, Z., Usandizaga, A.: Human-Centric Ontology-Based Context Modelling in Tourism. *In: Mediterranean Conference on Information Systems*, pp. 424-434 (2009)
[169] Hur, H., Park, H., Kim, J., Chang, H., Lee, M., Cho, S., Choi, E.: Context Model of User Aspect in Context-Awareness. *In: 10th IEEE International Conference on Advanced Communication Technology*, pp. 1647-1651 (2008)
[170] Coma, C., Cuppens-Boulahia, N., Cuppens, F., Cavalli, A.R.: Context Ontology for Secure Interoperability. *In: 3rd IEEE International Conference on Availability, Reliability and Security*, pp. 821-827 (2008)
[171] Tan, P.S., Goh, A.E., Lee, S.S.: A Context Model for B2B Collaborations. *In: IEEE International Conference on Services Computing*, pp. 108-115 (2008)
[172] Sun, L., Dai, P., Tao, L., Xu, G.: A Generic Context Information System for Intelligent Vision Applications. *In: 8th International Conference on Computational Science*, pp. 759-769 (2008)
[173] Ouyang, J.Q., Ding, B., Wang, H.M., Shi, D.X.: Component Based Context Model. *In: 9th IEEE International Conference on Web-Age Information Management*, pp. 569-574 (2008)
[174] Ko, K., Sim, K.: Development of Context Aware System Based on Bayesian Network Driven Context Reasoning Method and Ontology Context Modeling. *In: IEEE International Conference on Control, Automation and Systems*, pp. 2309-2313 (2008)

[175] Chang, H., Shin, S., Chung, C.: Modeling Context Life Cycle for Building Smarter Applications in Ubiquitous Computing Environments. *In: OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pp. 851-860 (2008)

[176] Ghadiri, N., Nematbakhsh, M.A., Baraani-Dastjerdi, A., Ghasem-Aghaee, N.: A Context-Aware Service Discovery Framework Based on Human Needs Model. *In: 5th International Conference on Service-Oriented Computing*, pp. 404-409 (2007)

[177] Erfianto, B., Mahmood, A.K., Rahman, A.S.: Modeling Context and Exchange Format for Context-Aware Computing. *In: 5th IEEE Student Conference on Research and Development*, pp. 1-5 (2007)

[178] Qin, W., Shi, Y., Suo, Y.: Ontology-Based Context-Aware Middleware for Smart Spaces. *Journal of Tsinghua Science & Technology*. 12, 707-713 (2007)

[179] Lee, D., Meier, R.: Primary-Context Model and Ontology: A Combined Approach for Pervasive Transportation Services. *In: 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 419-424 (2007)

[180] Mei, C., Tiexin, C., Hongqin, W.: A Context-Based Model of Knowledge Transferring in Project Management. *In: IEEE International Conference on Grey Systems and Intelligent Services*, pp. 1465-1469 (2007)

[181] Paganelli, F., Bianchi, G., Giuli, D.: A Context Model for Context-Aware System Design Towards the Ambient Intelligence Vision: Experiences in the e-Tourism Domain. *In: Universal access in ambient intelligence environments*, pp. 173-191 (2007)

[182] Hwang, Z., Uhm, Y., Lee, M., Kim, Y., Kim, G., Park, S.: A Context Model by Ontology and Rule for Offering the User-Centric Services in Ubiquitous Computing. *In: IEEE International Conference on Convergence Information Technology*, pp. 77-82 (2007)

[183] Choi, O., Yoon, Y.: A Meta Data Model of Context Information for Dynamic Service Adaptation on User Centric Environment. *In: IEEE International Conference on Multimedia and Ubiquitous Engineering*, pp. 108-113 (2007)

[184] Hwang, H.S., Shin, S.H., Kim, K.U., Lee, S.C., Kim, C.S.: A Context-aware System Architecture Using Personal Information Based on Ontology. *In: 5th IEEE ACIS International Conference on Software Engineering Research, Management & Applications*, pp. 610-615 (2007)

[185] Jrad, Z., Aufaure, M., Hadjouni, M.: A Contextual User Model for Web Personalization. *In: International Conference on Web Information Systems Engineering*, pp. 350-361 (2007)

[186] Pessoa, R., Calvi, C., Filho, J., Farias, C., Neisse, R.: Semantic Context Reasoning Using Ontology Based Models. *In: Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, pp. 44-51 (2007)

[187] Hamdeh, N., Ma, S.: OWL-Based Ontology for Secure and Adaptable Ubiquitous Environment. *In: 3rd IEEE International Conference on Semantics, Knowledge and Grid*, pp. 230-235 (2007)

[188] Strimpakou, M.A., Roussaki, I.G., Anagnostou, M.E.: A Context Ontology for Pervasive Service Provision. *In: 20th IEEE International Conference on Advanced Information Networking and Applications*, pp. 5 (2006)

[189] Sheng, F., Min, Z., Ling, H., Yongshan, W.: The Social Context Model for Research Activities in a Cyberspace. *In: 16th IEEE International Conference on Artificial Reality and Telexistence Workshops*, pp. 270-274 (2006)

[190] Wang, G., Jiang, J., Shi, M.: A Context Model for Collaborative Environment. *In: 10th IEEE International Conference on Computer Supported Cooperative Work in Design*, 1-6 (2006)

[191] Yang, S.J., Huang, A.F., Chen, R., Tseng, S.S., Shen, Y.S.: Context Model and Context Acquisition for Ubiquitous Content Access in u-learning Environments. *In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 78-83 (2006)

[192] Kranenburg, H.V., Bargh, M., Iacob, S., Peddemors, A.: A Context Management Framework for Supporting Context-Aware Distributed Applications. *Journal of IEEE Communications Magazine*. 44, 67-74 (2006)

[193] Chaari, T., Ejigu, D., Laforest, F., Scuturici, V.M.: Modeling and Using Context in Adapting Applications to Pervasive Environments. *In: ACS/IEEE International Conference on Pervasive Services*, pp. 111-120 (2006)
[194] Ou, S., Georgalas, N., Azmoodeh, M., Yang, K., Sun, X.: A Model Driven Integration Architecture for Ontology-Based Context Modeling and Context-Aware Application Development. *In: 2nd European Conference on Model Driven Architecture–Foundations and Applications*, pp. 188-197 (2006)
[195] Almeida, D.R., Baptista, C.S., da Silva, E.R., Campelo, C.E., Figueiredo, H.F., Lacerda, Y.A.: A Context-Aware System Based on Service-Oriented Architecture. *In: 20th IEEE International Conference on Advanced Information Networking and Applications*, pp. 6 (2006)
[196] Hu, B., Moore, P.: Context Modeling to Support Location Based Cooperative Mobile Learning. *In: 10th IEEE International Conference on Computer Supported Cooperative Work in Design*, pp. 1-5 (2006)
[197] Amundsen, S.L., Eliassen, F.: Combined Resource and Context Model for QoS-Aware Mobile Middleware. *In: 19th International Conference on Architecture of Computing Systems*, pp. 84-98 (2006)
[198] Hong, C.S., Kim, H., Kim, H.S., Lee, H.C.: An Approach for Configuring Ontology-Based Application Context Model. *In: ACS/IEEE International Conference on Pervasive Services*, pp. 337-340 (2006)
[199] Kalyan, A., Gopalan, S., Sridhar, V.: Hybrid Context Model Based on Multilevel Situation Theory and Ontology for Contact Centers. *In: 3rd IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 3-7 (2005)
[200] Go, Y.C., Sohn, J.C.: Context Modeling for Intelligent Robot Services Using Rule and Ontology. *In: 7th IEEE International Conference on Advanced Communication Technology*, pp. 813-816 (2005)
[201] Zacarias, M., Marques, A.R., Pinto, H.S., Tribolet, J.: Enhancing Collaboration Services with Business Context Models. In: *CEUR Proceedings of the Workshops on Cooperative Systems and Context*, pp. 1-10 (2005)
[202] Ko, E.J., Lee, H.J., Lee, J.W.: An Intelligent Context-Aware Service Engine Based on Ontology. *In: IEEE International Conference on Computational Intelligence for Modeling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, pp. 485-491 (2005)
[203] Kaenampornpan, M., O'Neill, E.: An Integrated Context Model: Bringing Activity to Context. *In: Workshop on Advanced Context Modelling, Reasoning and Management* (2004)
[204] Miller, G.A.: WordNet: A Lexical Database for English. *Journal of Communications of the ACM*. 38, 39-41 (1995)
[205] Oriol, M., Marco, J., Franch, X.: Quality Models for Web Services: A Systematic Mapping. *Journal of Information and Software Technology*. 56, 1167-1182 (2014)
[206] Ameller, D., Burgués, X., Collell, O., Costal, D., Franch, X., Papazoglou, M.: Development of Service-Oriented Architectures Using Model-Driven Development: A Mapping Study. *Journal of Information and Software Technology*. 62, 42-66 (2015)
[207] Šmite, D., Wohlin, C., Gorschek, T., Feldt, R.: Empirical Evidence in Global Software Engineering: A Systematic Review. *Journal of Empirical Software Engineering*. 15, 91-118 (2010)
[208] Cavalcante, E., Pereira, J., Alves, M., Maia, P., Moura, R., Batista, T., Delicato, F., Pires, P.: On the Interplay of Internet of Things and Cloud Computing: A Systematic Mapping Study. *Journal of Computer Communications*. 89, 17-33 (2016)
[209] Uschold, M., King, M.: Towards a Methodology for Building Ontologies. *In: Workshop on Basic Ontological Issues in Knowledge Sharing* (1995)

[210] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. *In: Computer Science Series*. Springer (2012)

[211] Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. *Journal of Knowledge Acquisition*. 5, 199-220 (1993)

[212] Borst, W.N.: Construction of Engineering Ontologies for Knowledge Sharing and Reuse. *Ph.D. Thesis*, Twente (1997)

[213] Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, Tools and Languages for Building Ontologies. Where is Their Meeting Point?. *Journal of Data & Knowledge Engineering*. 46, 41-64 (2003)

[214] Guarino, N.: Formal Ontology in Information Systems. *In: 1rst International Conference on Formal Ontology in Information Systems*, pp. 3-15 (1998)

[215] Grüninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. *In: Workshop on Basic Ontological Issues in Knowledge Sharing* (1995)

[216] Staab, S., Studer, R., Schnurr, H.P., Sure, Y.: Knowledge Processes and Ontologies. *In: IEEE intelligent Systems*, pp. 26-34 (2001)

[217] Pinto, H.S., Gómez-Pérez, A., Martins, J.P.: Some Issues on Ontology Integration. *In: Workshop on Ontologies and Problem Solving Methods* (1999)

[218] Pinto, H.S., Martins, J.P.: A Methodology for Ontology Integration. *In: 1st ACM International Conference on Knowledge Capture*, pp. 131-138 (2001)

[219] Papazoglou M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. *In: 4th IEEE International Conference on Web Information Systems Engineering*, pp. 3-12 (2003)

[220] Su, X., Ilebrekke, L.: A Comparative Study of Ontology Languages and Tools. *In: 14th International Conference on Advanced Information Systems Engineering*, pp. 761-765 (2002)

[221] Hella, L., Krogstie, J.: A Structured Evaluation to Assess the Reusability of Models of User Profiles. *In: 11th International Workshop on Enterprise, Business-Process and Information Systems Modeling*, pp. 220-233 (2010)

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

[223] Gómez-Pérez, A.: Evaluation of Ontologies. *Journal of Intelligent Systems*. 16, 391-409 (2001)

[224] Kishore, R., Sharman, R.: Computational Ontologies and Information Systems I: Foundations. *Journal of Communications of the Association for Information Systems*. 14, 8 (2004)

[225] Hu, B., Moore, P., Chen, H.: A Semantic Context Model for Location-Based Cooperative Mobile Computing. *In: IEEE International Conference on Communications*, pp. 326-331 (2007)

[226] Paganelli, F., Giuli, D.: An Ontology-Based Context Model for Home Health Monitoring and Alerting in Chronic Patient Care Networks. *In: 21st IEEE International Conference on Advanced Information Networking and Applications Workshops*, pp. 838-845 (2007)

[227] Kayes, A.S.M., Han, J., Colman, A.: OntCAAC: An Ontology-Based Approach to Context-Aware Access Control for Software Services. *The Computer Journal*. 58, 3000-3034 (2015)

[228] Kayes, A.S.M., Han, J., Colman, A.: An Ontological Framework for Situation-Aware Access Control of Software Services. *Journal of Information Systems*. 53, 253-277 (2015)

[229] Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: An Ontology-Based Context Model in Intelligent Environments. *In: International Conference of Communication networks and distributed systems modeling and simulation*, pp. 270-275 (2004)

[230] Xiong, Z., Dixit, V.V., Waller, S.T.: The Development of an Ontology for Driving Context Modelling and Reasoning. *In: 19th IEEE International Conference on Intelligent Transportation Systems*, pp. 13-18 (2016)

[231] Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D.: A Survey of Semantics-Based Approaches for Context Reasoning in Ambient Intelligence. *In: Workshop on Constructing ambient intelligence*, pp. 14-23 (2007)
[232] Parent, C., Spaccapietra, S.: An Overview of Modularity. *In: Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.) Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. Lecture Notes in Computer Science, vol. 5445*. Springer 5-23 (2009)
[233] Ji, Q., Haase, P., Qi, G., Hitzler, P., Stadtmüller, S.: RaDON-Repair and Diagnosis in Ontology Networks. *In: European Semantic Web Conference*, pp. 863-867 (2009)
[234] Danila, C., Stegaru, G., Stanescu, A.M., Serbanescu, C.: Web-Service Based Architecture to Support SCM Context-Awareness and Interoperability. *Journal of Intelligent Manufacturing*. 27, 73-82 (2016)
[235] Lee, Y., Iyengar, S.S., Min, C., Ju, Y., Kang, S., Park, T., Lee, J., Rhee, Y. Song, J.: Mobicon: A Mobile Context-Monitoring Platform. *Journal of Communications of the ACM*. 55, 54-65 (2012)
[236] Bettini, C., Maggiorini, D., Riboni, D.: Distributed Context Monitoring for the Adaptation of Continuous Services. *Journal of World Wide Web Internet and Web Information Systems*. 10, 503-528 (2007)
[237] Kang, S., Lee, Y., Min, C., Ju, Y., Park, T., Lee, J., Rhee, Y., Song, J.: Orchestrator: An Active Resource Orchestration Framework for Mobile Context Monitoring in Sensor-Rich Mobile Environments. *In: IEEE International Conference on Pervasive Computing and Communications*, pp. 135-144 (2010)
[238] Kang, S., Lee, J., Jang, H., Lee, Y., Park, S., Song, J.: A Scalable and Energy-Efficient Context Monitoring Framework for Mobile Personal Sensor Networks. *Journal of IEEE Transactions on Mobile Computing*. 9, 686-702 (2010)
[239] Mitchell, M., Meyers, C., Wang, A.I.A., Tyson, G.: Contextprovider: Context Awareness for Medical Monitoring Applications. *In: International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5244-5247 (2011)
[240] Lee, Y., Ju, Y., Min, C., Kang, S., Hwang, I., Song, J.: Comon: Cooperative Ambience Monitoring Platform with Continuity and Benefit Awareness. *In: 10th ACM International Conference on Mobile Systems, Applications, and Services*, pp. 43-56 (2012)
[241] Cho, K., Hwang, I., Kang, S., Kim, B., Lee, J., Lee, S., Park, S., Song, J., Rhee, Y.: HiCon: A Hierarchical Context Monitoring and Composition Framework for Next-Generation Context-Aware Services. *Journal of IEEE Network*. 22, 34-42 (2008)
[242] Murao, K., Terada, T., Takegawa, Y., Nishio, S.: A Context-Aware System that Changes Sensor Combinations Considering Energy Consumption. *In: 6th International Conference on Pervasive Computing*, pp. 197-212 (2008)
[243] Riva, O.: Contory: A Middleware for the Provisioning of Context Information on Smart Phones. *In: International Conference on Middleware*, pp. 219-239 (2006)
[244] Stal, M.: Using Architectural Patterns and Blueprints for Service-Oriented Architecture. *Journal of IEEE Software*. 23, 54-61 (2006)
[245] Nagios, The Industry Standard in IT Infrastructure Monitoring, https://www.nagios.org/
[246] Franch, X., Carvallo, J.: Using Quality Models in Software Package Selection. *Journal of IEEE Software*. 20, 34-41 (2003)
[247] International Organization for Standardization, ISO/IEC standard 9126: Software Engineering – Product Quality, part 1 (2001)
[248] Richardson, L., Ruby, S.: RESTful web services. O'Reilly Media, Sebastopol (2008)
[249] Twitter API, https://dev.twitter.com/
[250] SocialMention, Real-Time Social Media Search and Analysis, http://socialmention.com/
[251] Apache Kafka, A High-Throughput Distributed Messaging System, http://kafka.apache.org/
[252] Google Analytics, https://analytics.google.com/
[253] Cabrera, O., Oriol, M., Franch, F., Marco, J.: Online Annexes of CAMA: A Context-Aware Monitoring Architecture for Highly Dynamic Environments. http://gessi.lsi.upc.edu/CAMAannexes/

[254] Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is Key. *Journal of Communications of the ACM*. 48, 49-53 (2005)

[255] Kofod-Petersen, A., Aamodt, A.: Contextualised ambient intelligence through case-based reasoning. *In: European Conference on Case-Based Reasoning*, pp. 211-225 (2006)

[256] Goker, A., Davis, J.: Information Retrieval: Searching in the 21st Century. *In: Technology and Engineering*. Wiley (2009)