A cotutelle Thesis
Submitted in fulfilment of the requirements for the
Doctor Degree in Computer Science with International Mention
At the Universitat Politècnica de València and the University of Sousse

Towards an Approach for Intelligent Adaptation Decision-Making of Pervasive Middleware

Publicly defended on the 11th of January 2023
By

Roua Jabla

Supervisors

Félix Buendia Garcia, Professor	Universitat Politècnica de València, Spain
Sami Faiz, Professor	University of Tunis, Tunisia
Maha Khemaja, Associate Professor	University of Sousse, Tunisia

External Evaluators

Lotfi Ben Romdhane, Professor	University of Sousse, Tunisia
Juan Manuel Corchado Rodriguez, Professor	University of Salamanca, Spain
Juan Carlos Nieves Sánchez, Professor	University of Umeå, Sweden

Committee Members

Habib Youssef, Professor	University of Sousse, Tunisia	President
Miguel Ángel Sicilia Urbán, Professor	University of Alcalá, Spain	Secretary
Ahmed Hadj Kacem, Professor	University of Sfax, Tunisia	Vocal

ABSTRACT

This thesis describes research to gain insight into pervasive middleware solutions and context-aware solutions that expand their perspective from static to dynamic pervasive environments. The motivation behind this research arose from a need to reconsider and replace today's context-aware solutions with more intelligent solutions to account for dynamic environments and users' preferences changes at runtime. In this context, the end goal is to focus on offering intelligent context-aware solutions that could deal with the automatic context model evolution and new decisions generation according to context changes at runtime. To do so, in the current thesis, we illustrate a hybrid approach termed IConAS - a means of combining the practical advantages of context evolution with the decision-making adaptation. This combination leads to intelligent context-aware solutions that could reflect changes occurring in their surrounding dynamic environments at runtime.

The thesis concentrates on the three important contributions as follows:

- Definition of the IConAS approach that combines two main approaches. This hybrid approach aims to offer intelligent context-aware solutions through augmenting an existing middleware. The purpose of this augmentation is to support runtime and automatic context evolution and decision-making adaptation in order to reflect changes in dynamic environments;
- Introduction of the first part of our hybrid approach: the CoE approach. This approach aims to establish an ontology-based context model evolution based on an unsupervised ontology learning approach. Therefore, it automatically evolves an ontology-based context model according to context changes occurring in surrounding dynamic environments at runtime;
- Introduction of the second part of our hybrid approach: the DMA approach. This approach aims to automatically learn and generate decision rules and subsequently, enrich a rules knowledge base at runtime to cope with changes and evolved ontology-based context models. It is relying on the use of Machine Learning and a Genetic Algorithm.

These contributions are validated through different perspectives:

- First, the evaluation of the CoE approach is performed using feature-based, criteria-based, expert-based and competency question-based evaluation approaches;
- Second, the evaluation of the DMA approach is established through assessing its effectiveness in terms of number of rules, performance and computational time;

- Finally, the evaluation of the IConAS approach is conducted through an elderly healthcare case study together with activity recognition and user satisfaction evaluation approaches.

Resumen

Esta tesis describe la investigación para obtener información sobre soluciones de middleware y soluciones sensibles al contexto que amplían la perspectiva de entornos estáticos a entornos dinámicos generalizados. La motivación detrás de esta investigación surgió de la necesidad de reconsiderar y reemplazar las soluciones sensibles al contexto actuales con soluciones más inteligentes para dar cuenta de los entornos dinámicos y los cambios de preferencias de los usuarios en el tiempo de ejecución. En este sentido, el objetivo final es centrarse en ofrecer soluciones inteligentes sensibles al contexto que puedan abordar la evolución automática del modelo de contexto y la generación de nuevas decisiones de acuerdo con los cambios de contexto en tiempo de ejecución. Con este fin, en la tesis actual ilustramos un enfoque híbrido denominado IConAS, que combina las ventajas prácticas de la evolución del contexto con la adaptación en la toma de decisiones. Esta combinación conduce a soluciones inteligentes sensibles al contexto que podrían reflejar los cambios que ocurren en sus entornos dinámicos en tiempo de ejecución.

La tesis se concentra en las tres contribuciones importantes de la siguiente manera:

- Definición del enfoque IConAS que combina dos enfoques principales. Este enfoque híbrido tiene como objetivo ofrecer soluciones inteligentes sensibles al contexto mediante la extensión de una solución middleware existente. El propósito de esta extensión consiste en dar soporte en tiempo de ejecución a la evolución automática del contexto y la adaptación de la toma de decisiones para reflejar los cambios en entornos dinámicos;

- Introducción de la primera parte de nuestro enfoque híbrido: el enfoque CoE. Este enfoque tiene como objetivo establecer una evolución de modelo de contexto a partir de una ontología basada en un enfoque de aprendizaje no supervisado. Por lo tanto, desarrolla automáticamente un modelo de contexto basado en dicha ontología de acuerdo con los cambios de contexto que ocurren en los entornos dinámicos en tiempo de ejecución;

- Introducción de la segunda parte de nuestro enfoque híbrido: el enfoque DMA. Este enfoque tiene como objetivo aprender y generar automáticamente reglas de decisión y, posteriormente, enriquecer una base de conocimientos de reglas en tiempo de ejecución para hacer frente a los cambios y modelos de contexto basados en modelos de ontología evolucionados. Se basa en el uso de técnicas de Machine Learning y el uso de un Algoritmo Genético.

Estas contribuciones se validan desde diferentes perspectivas:

- Primero, la evaluación del enfoque CoE se realiza utilizando enfoques de evaluación basados en características, criterios, expertos y preguntas de competencia;
- En segundo lugar, la evaluación del enfoque DMA se establece evaluando su eficacia en términos de número de reglas, rendimiento y tiempo computacional;
- Finalmente, la evaluación del enfoque IConAS se lleva a cabo a través de un estudio de caso de atención médica para personas mayores junto con enfoques de reconocimiento de actividad y evaluación de la satisfacción del usuario.

Resum

Aquesta tesi descriu la recerca per obtenir informació sobre solucions middleware i solucions sensibles al context que amplien la perspectiva d'entorns estàtics a entorns dinàmics generalitzats. La motivació darrere aquesta investigació va sorgir de la necessitat de reconsiderar i reemplaçar les solucions sensibles al context actuals amb solucions més intel·ligents per donar compte dels entorns dinàmics i els canvis de preferències dels usuaris en el temps d'execució. En aquest sentit, l'objectiu final es centrar en oferir solucions intel·ligents sensibles al context que puguin abordar l'evolució automàtica del model de context i la generació de noves decisions d'acord amb els canvis de context en temps d'execució. Amb aquesta finalitat, a la tesi actual il·lustrem un enfocament híbrid anomenat IConAS, que combina els avantatges pràctics de l'evolució del context amb l'adaptació a la presa de decisions. Aquesta combinació condueix a solucions intel·ligents sensibles al context que podrien reflectir els canvis que tenen lloc als seus entorns dinàmics en temps d'execució.

La tesi es concentra en les tres contribucions importants de la manera següent:

- Definició de l'enfocament IConAS que combina dos aspectes principals. Aquest enfocament híbrid té com a objectiu oferir solucions intel·ligents sensibles al context mitjançant l'extensió d'una solució middleware existent. El propòsit d'aquesta extensió consisteix a donar suport en temps d'execució a l'evolució automàtica del context l'adaptació de la presa de decisions per reflectir els canvis en entorns dinàmics;

- Introducció de la primera part del nostre enfocament híbrid: enfocament CoE. Aquest enfocament té com a objectiu establir una evolució de model de context a partir duna ontologia basada en un enfocament d'aprenentatge no supervisat. Per tant, desenvolupa automàticament un model de context basat en aquesta ontologia d'acord amb els canvis de context que ocorren en els entorns dinàmics en temps d'execució;

- Introducció de la segona part del nostre enfocament híbrid: enfocament DMA. Aquest enfocament té com a objectiu aprendre i generar automàticament regles de decisió i, posteriorment, enriquir una base de coneixements de regles en temps d'execució per fer front als canvis i models de context basats en models d'ontologia evolucionats. Es basa en l'ús de tècniques de Machine Learning i l'ús d'un algoritme genètic.

Aquestes contribucions es validen des de diferents perspectives:

- Primer, l'avaluació de l'enfocament CoE es realitza utilitzant tècniques d'avaluació basades en característiques, criteris, experts i preguntes de competència;

- En segon lloc, l'avaluació de l'enfocament DMA s'estableix avaluant la seva eficàcia en termes de nombre de regles, rendiment i temps computacional;
- Finalment, l'avaluació de l'enfocament IConAS es duu a terme a través d'un estudi de cas d'atenció mèdica per a gent gran juntament amb enfocaments de reconeixement d'activitat i avaluació de la satisfacció de l'usuari.

ACKNOWLEDGEMENTS

Foremost, I would like to express my deepest gratitude to my thesis supervisors Prof. *Félix Buendia Garcia* and Prof. *Sami Faiz* for their constant support and guidance throughout the whole process of research and writing this thesis.

Next, I would like to extend my heartiest gratitude and thanks to my co-supervisor Prof. *Maha Khemaja*, who believed in me from the beginning of this thesis and who never ceased to encourage, guide and kindly nudge me forward when I doubt my ability or could not see the finish line. This thesis would not have been achievable without her constant feedback, her enormous research contributions and her emotional support.

I am sincerely thankful to my external evaluators of the thesis, Prof. *Lotfi Ben Romdhane*, Prof. *Juan Manuel Corchado Rodriguez* and Prof. *Juan Carlos Nieves Sánchez* for their positive feedback and their additional constructive comments.

A special thanks goes out to the members of my defense committee: Prof. *Habib Youssef*, Prof. *Miguel Ángel Sicilia Urbán* and Prof. *Ahmed Hadj Kacem*. It is a great honor to have you on this committee.

A warm thanks from the bottom of my heart to my parents, my brother and my sister who have given me unconditional support throughout all my years of education and who supported me emotionally, spiritually and financially throughout this work. I shall always be indebted to them.

Last, but not least, my appreciation goes to my friend and colleague Amani for her kind support throughout the thesis process. Thank you for your tremendous support during my crucial times and your interesting discussions that always inspired me and increased my motivation.

Contents

Acronyms

BFF Backend-For-Frontend. 130–132

CAMPUS Context-Aware Middleware for Pervasive and Ubiquitous Service. 26

CARISMA Context-aware Reflective Middleware System for Mobile Applications. 23

CoBrA Context Broker Architecture. 24

CoE Context Evolution. 46, 61

CQs Competency Questions. 114

CSQ-8 8-item Client Satisfaction Questionnaire. 143

DMA Decision-Making Adaptation. 46, 89

DSL Domain Specific Language. 97

DT Decision Table. 65, 124

EC Engineering Cycle. 5

extJWNL extended Java WordNet Library. 74

FN False Negative. 123, 124, 137

FP False Positive. 123, 124, 137, 138

GA Genetic Algorithm. 41, 90

HAR Human Activity Recognition. 49

HTML Hypertext Markup Language. 14

IDE Integrated Development Environment. 131

IntElyCare Intelligent Elderly Healthcare. 48–50, 130

IR Inheritance Richness. 109

IRR Inverse Relations Ratio. 109

LSPs Lexico-Syntactic Patterns. 66

MADAM Mobility and ADaptation enAbling Middleware. 25

MUSIC Mobile USers In Ubiquitous Computing. 25

NB Naïve Bayes. 41

OOPS! Ontology Pitfall Scanner. 108

OWL Web Ontology Language. 16, 62

PARC Palo Alto Research Center. 11

POS Parts of Speech. 67

RDF Resource Description Framework. 16

RDFS Resource Description Framework Schema. 16

RR Relationship Richness. 109

SGML Standard Markup Language. 14

SOA Service Oriented Architectures. 25

TF-IDF Term Frequency/Inverted Document Frequency. 64

TN True Negative. 123, 124, 137

TP True Positive. 123, 124, 137, 138

UI User Interface. 132

UML Unified Modeling Language. 14

XML eXtensible Markup Language. 14, 62, 64, 65

XSD XML Schema Definition. 64

XSOM XML-Schema Object Model. 74

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

1.1 Motivation and Problem Statement

Two decades ago, Mark Weiser [Weiser, 1991, Weiser and Brown, 1997] envisioned that our lives will be evolving around pervasive computing technology, which is an attractive vision in the world of computing. The rationale behind pervasive computing is to reach the idea of providing users with seamless access to the information everywhere and at any time through small devices embedded in their surrounding environments. Weiser's vision, with its focus on users and their tasks rather than on devices and technology, had inspired a generation of researchers in creating a new shift from "one person–one desktop computer" to smaller devices that are spreading ubiquitously and invisibly throughout the environment while providing endless support to users [Weiser et al., 1999].

Now that we are firmly in the 21st century and beyond millennialism, we have seen the rise of Weiser's vision [Cook and Das, 2007, Conti et al., 2012]. Therefore, pervasive computing becomes one of the most growing eras of computing, where traditional boundaries between users, devices, applications and information get more and more blurred in light of the shift exhibited in Weiser's vision. Additionally, there is a high attention on context-aware computing expressed by the pervasive computing community, which considers context as a key field to bridge the physical and the virtual worlds and to perceive context information and users' preferences. Hence, context-aware computing grasps context knowledge, such as where we are and what we are doing, in order to make better decisions about the enclosing environment and to provide what we need. The convergence of pervasive computing together with context-aware computing has witnessed a significant growth in interest in the different pervasive middleware solutions and context-aware solutions, such as systems, frameworks or applications. These solutions focus on context reasoning to deal with high-level inference of pervasive contextual information, to tailor application behavior and to trigger adaptation based on the changing context as well as users' preferences at runtime.

Apart from that, with the advancement of technology in the 2010s, mobile devices came along and users became increasingly mobile. The wireless nature of these mobile devices has fostered a new era of mobility since they are able to arbitrarily join and leave an environment and to be transported to anywhere at any time and by any user. The sustained growth of mobile devices supports Weiser's vision and reaches its intersection with pervasive computing environments. In line with this intersection, middleware and context-aware solutions expand their perspective from static to dynamic environments. Consequently, context becomes extremely rich and changing on account of the dynamic nature of users' surrounding environments and the users' preferences. Stated differently, due to users' mobility and environment dynamicity, existing solutions could face emerging changes in their enclosing environments or in users' preferences and behaviors over time. Accordingly, they may reflect inaccuracies due to the fact that they are usually focused on static and predictable environments with known users' behaviors and preferences. Therefore, these solutions are powerless to deal with unforeseen and new changes occurring in enclosing dynamic environments or in their users' preferences. A major issue currently facing such kinds of solutions is the fact that, obviously, they depend on static context models that are defined during the design phase and remain the same overtime considering only context information changes at runtime without considering evolution and learning foundations [Hoareau and Satoh, 2009] to reflect dynamic environments' changes at runtime. Hence, any additional context feature (i.e., entity) thrown in by a new environment change or new users' preference at runtime would have to be developed from scratch based on human contributions. The adoption of existing solutions, however, is limited because they are restricted to predictable environments and provide context models that are static environment-dependent. These constitute the main limits for addressing efficiently context-awareness and dynamic environments within existing solutions that could suffer from inaccuracy or interruption at runtime and require human contributions.

Another serious issue is that almost existing middleware and context-aware solutions follow a static decision-making process based on predefined decision rules, despite the high changes in the enclosing dynamic environments and users' preferences. Disregarding users' mobility and environment dynamicity, existing solutions usually work well with a static decision-making process that can only react to changes in environment attributes and context information [Cook and Das, 2004]. However, they can handle neither dynamic environment nor context changes at runtime that yield poor performance [Cook and Das, 2004] and results in overdetermined and unrealistic decisions for users at runtime. On that account, existing solutions are considered as not well-suited to the dynamic nature of environments since unforeseen and new changes could affect the quality of decision-making process and decisions being made at runtime. Consequently, it is certain that a static decision-making process with a priori defined rules could lose its efficiency in dynamic environments.

For these aforementioned issues, today's middleware and context-aware solutions in dynamic environments should be reconsidered and replaced with more intelligent solutions to account for dynamic environments and users' preferences changes at runtime. In this direction, two fundamental challenges emerge and need to be addressed to solve these issues faced in traditional and existing solutions:

- First, context models need to be automatically extended when users' preferences or the running environment suffer from a change at runtime, as a consequence of addition of new environment attributes or mobile device migration. Indeed, these models shall be updated to cope with highly dynamic environments and changing users' preferences and behaviors. Therefore, context-aware solutions must support seamless and automatic continuous evolution on their context models when environment changes are met or a user moves to another environment at runtime.

- Second, a grand challenge is the need to timely adjust context-aware solutions' behaviors to the dynamics entailed in their surrounding environments or in their users' preferences and behaviors at runtime. To set up a dynamic environment, solutions should reflect dynamic changes at runtime through extending theory regarding static decision-making to bridge the gap between the predefined decision rules and the new changes. Therefore, a considerable need for a decision-making process adaptation, which aims to continuously learn new decision rules, has emerged to make solutions more effective in dynamic environments at runtime.

1.2 Thesis Objectives and Questions

Based upon the predominating motivations and challenges, the research heart and the main objective <***O***> of this thesis is to propose a hybrid approach that aims to augment an existing middleware solution for providing intelligent context-aware solutions. This proposed approach offers context-aware solutions the ability to automatically and dynamically evolve their context models and their decision-making when facing changes in enclosing dynamic environments or in users' preferences and behaviors at runtime, without lowest human contribution possible.

The fulfillment of this main objective <***O***> lies in two research sub-objectives in this thesis as follows:

- The first sub-objective <***O.1***> is to propose an approach to automatically evolve and extend context models at runtime to answer users' mobility and frequent changes in the surroundings. In this approach, the context model requires an automatic involvement of newly available and unforeseen context features and relationships between them appearing in the surroundings at runtime. Therefore, context evolution enables a context-aware solution to operate in dynamic environments in order to decrease uncertainty and to effectively reflect the different kinds of changes occurring at runtime. This context evolution, particularly, would fundamentally affect the quality of decision-making process and decisions being made for users at runtime, which leads to the second sub-objective of this thesis;

- The second objective <***O.2***> is to provide another approach to automatically adjust behaviors of context-aware solutions in the wake of changes entailed in their surroundings as well as their context models at runtime. This approach aims to consider decision-making adaptation and decision rule learning to automatically learn and generate new decision rules at runtime. Therefore, the predefined decision rules at design time will be refined in-time to catch up with the rapid and relevant changes in dynamic environments and to avoid unnecessary runtime errors and users' dissatisfactions.

To achieve the main objective <***O***>, this thesis addresses a central research question:

- ***RQ*** *How can existing middleware solution boundaries be augmented to automatically meet changes in users' preferences and behaviors or in surrounding dynamic environments at runtime without the lowest human contribution possible?*

In order to find an answer to this broad research question ***RQ***, it is necessary to break it into subsidiary questions that are chosen as more concrete. The corresponding sub-questions ***RQ.1.*** and ***RQ.2.*** related to the sub-objectives ***<O.1>*** and ***<O.2>***, respectively, are:

- ***RQ.1.*** *How can the context evolution affected by users' preference and behaviors or enclosing environments' changes be automatically and dynamically supported at runtime?*
  - ***RQ.1.1.*** *What kind of context modeling approach should be followed to meet an evolution of context models at runtime?*
  - ***RQ.1.2.*** *How to take advantage of external knowledge bases to evolve context models?*
  - ***RQ.1.3.*** *How is the evolved context model effective?*
- ***RQ.2.*** *How can a decision-making process be adapted in an automatic manner at runtime?*
  - ***RQ.2.1.*** *How can we automatically enrich rule knowledge bases with missing decision rules and keep them up-to-date?*
  - ***RQ.2.2.*** *How can generated rules be used to improve the decision-making capabilities?*

1.3 Thesis Contributions

By answering the stated research questions, this thesis makes a key contribution:

A Hybrid Approach. We present a Hybrid approach, called Intelligent Context-Aware Solution (IConAS), that includes a set of two approaches to augment an existing middleware solution. This approach is intended to offer intelligent context-aware solutions that could support automatic context evolution and decision-making adaptation at runtime through involving an undefined set of context features accompanied with an undefined set of decision rules. It is presented to account for the inaccuracies brought by users' mobility and dynamic environments without compromising human interventions at runtime.

On this basis, this key contribution leads to sub-contributions that fall into the following categories:

- ***A Context Evolution Approach.*** We provide an approach, called Context Evolution (CoE), for automatically evolving context models according to changes detected in the enclosing environments or in users'preferences at runtime. In this approach, a context evolution architecture was conceived, which provides an unsupervised ontology learning approach for the particular case of ontology-based context modeling. The aim of this approach is two-fold: i) to analyze heterogeneous semi-structured input data for learning an ontology and ii) to use the learned ontology for extending and improving an existing ontology-based context model to accommodate the changes that occur in its closed dynamic environment.

- ***A Decision-Making Adaptation Approach.*** We propose an approach, called Decision-Making Adaptation (DMA), for automatically carrying out dynamic changes in context-aware solutions' behaviors through learning and generating news decision rules to answer relevant dynamics entailed in their surrounding environments at runtime. Therefore, this approach aims to continuously enriching a rule knowledge base through the automatic generation of efficient decision rules over time to make solutions more resilient to dynamic environments, context model evolution as well at runtime. It allows decisions to be more accurate, precise and personalized to satisfy non-predetermined changes.

1.4 Research Methodology

To reach the research objectives described in the previous section, applying a research methodology is necessary to assess whether derived results and conclusions are scientifically relevant [Bhattacherjee, 2012]. In this respect, the following subsections detail the research methodology followed in this thesis with specific research methods and techniques. First, a full explanation of the Design Science Research Methodology (DSRM) is given. Second, the decision to follow the DSRM is elaborated, entailed by an explanation of its activities.

1.4.1 Design Science Research Methodology

Design science [Simon, 1996, March and Smith, 1995, Hevner et al., 2004] is a Research methodology that fosters the design and investigation of artefacts that could be a model, a method, a framework, a process, a piece of software and so on, to solve problems of the environment and to contribute to the current knowledge base. In his approach, Wieringa [Wieringa, 2009, Wieringa, 2014] regards DSRM as a set of nested regulative cycles. Therefore, this methodology results in a two-cycle approach:

- The first cycle is the Engineering Cycle (EC) in which the artifact is investigated, designed and validated. The EC addresses the problem investigation, solution design, solution validation and solution implementation activities during its first iteration. Any subsequent iterations address the solution evaluation and improvement of the implementation. The EC is shown in Figure 1.1.

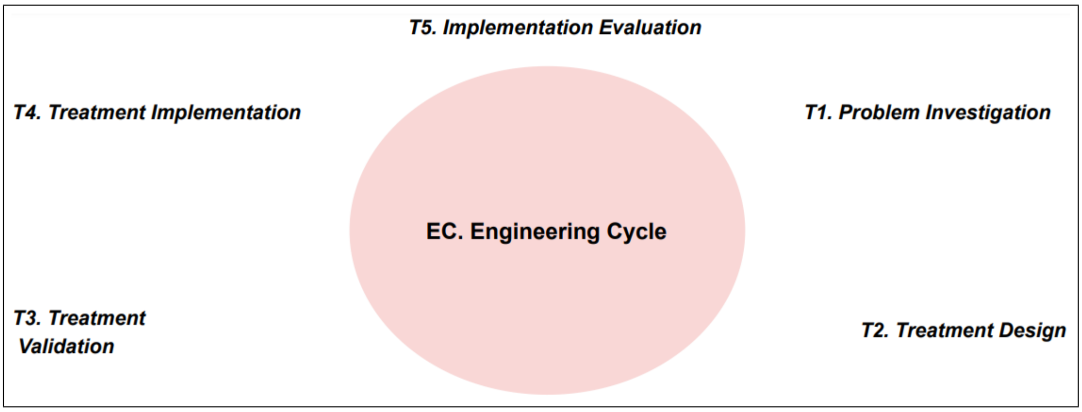

Figure 1.1: EC according to Wieringa.

As shown in Figure 1.1, the EC consists of the following steps:

  - ***Problem Investigation (T1).*** Before designing a solution, it is better to understand the problem. The goal of the Problem investigation step is to identify the problem and goals.
  - ***Treatment Design (T2).*** In this step, available treatments are identified and a treatment is created.
  - ***Treatment Validation (T3).*** The created treatment is validated to determine if it solves the identified problem.
  - ***Treatment Implementation (T4).*** The validated treatment is implemented in a real-life situation.
  - ***Treatment Evaluation/Problem Investigation (T5).*** After the implementation, the treatment is evaluated in the original context. If needed, this also serves as the problem investigation for the next cycle..

- The second cycle is the Empirical Research Cycle (ERC) that is defined as a rational way of answering scientific knowledge questions. The ERC addresses the research problem investigation, research design, design validation, research and result analysis as illustrated in Figure 1.2.

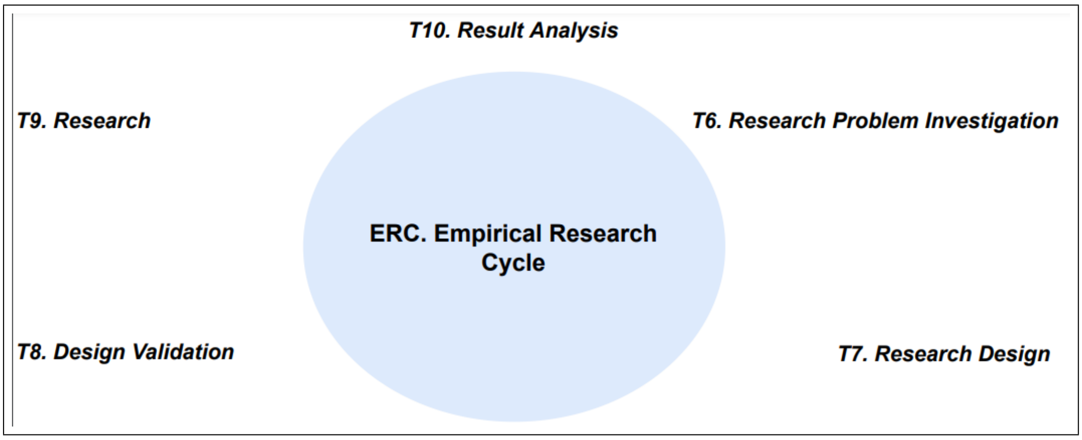

Figure 1.2: ERC according to Wieringa.

Thus, the DSRM process iterates between these both regulative cycles of designing an artefact to fulfill the desired need and the empirical investigation of this designed artefact within the problem context it was designed for.

1.4.2 Thesis Research Methodology

The basis of this thesis is embodied by the DSRM of Wieringa's approach [Wieringa, 2009, Wieringa, 2014]. For sake of brevity, we do not present all the iterations that we performed over the tasks of the DSRM cycles; instead, we describe the activities of designing and investigating by means of five tasks (T) that are part of an engineering cycle: T1 Problem investigation, T2 Treatment design, T3 Treatment validation, T4 Treatment implementation and T5 Implementation evaluation. Since the development of this thesis is in the frame of a DSRM, Figure 1.3 gives an overview of our DSRM, which includes the different activities performed in this thesis and the indication of the chapters that detail the engineering cycle's activities.

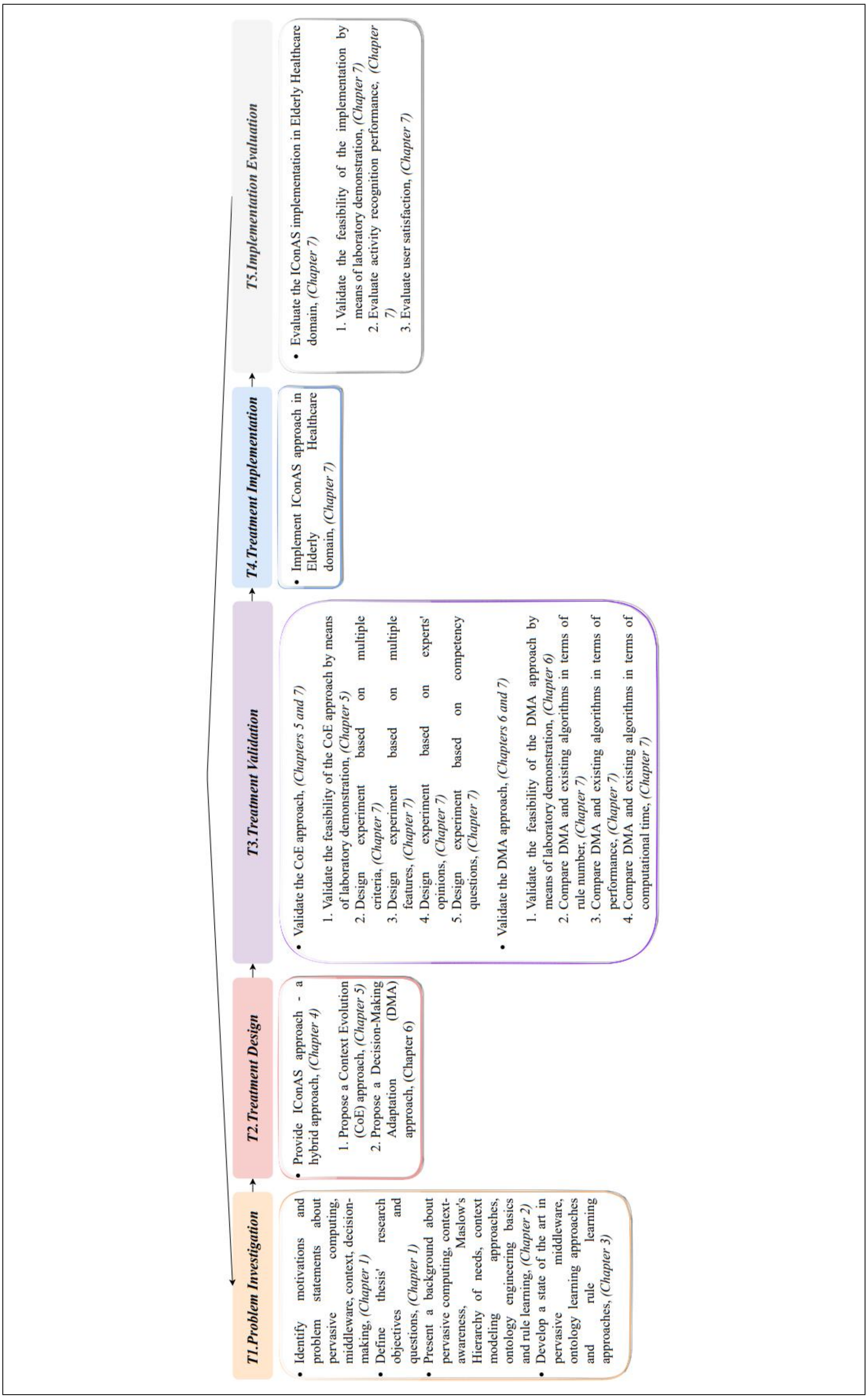

Figure 1.3: Thesis's DSRM.

As shown in Figure 1.3, our engineering cycle starts with the ***Problem Investigation (T1)*** where its main related tasks are:

1. the identification of thesis motivation and problems statements,
2. the definition of thesis research objectives and questions,
3. the presentation of a considerable background about pervasive computing, context-awareness, Maslow's Hierarchy of needs, context modeling approaches, ontology engineering basics and rule learning,
4. the development of a state-of-the-art in pervasive middleware, ontology learning approaches and rule learning approaches.

Then, this state-of-the-art reveals that existing pervasive middleware cannot automatically deal with dynamic environments at runtime and motivates the need for context evolution and dynamic decision-making adaptation at runtime to augment existing middleware and offer intelligent context-aware solutions. On this basis, the ***Treatment Design (T2)*** requires, first, to propose IConAS approach - a hybrid approach to reflect real-time contextual changes in dynamic environments. An initial attempt to propose IConAS approach aims at providing:

- First, CoE approach to evolve a context model at runtime according to changes in dynamic environments,
- Second, DMA approach to support dynamic decision-making through automatically learning, generating new decision rules and then enriching a rule knowledge base at runtime.

Afterwards, we validated the automatic ontology-based context evolution and the dynamic decision-making adaptation approaches. For the activities related to the ***Treatment Validation (T3)***, we followed the empirical research structure of the empirical cycle, so that we skipped the details about its possible activities for the sake of brevity. In general, different validation methods have been applied. For example, for the CoE approach, we decided to perform a laboratory demonstration, which illustrates the use of the CoE approach and exemplifies how it can be used, and to design experiments based on multiple criteria, features, experts' opinions and competency questions. And for DMA approach, we decided to ensure in several ways: a laboratory demonstration, which illustrates the use of the DMA approach, and a comparative experiment that assesses DMA approach with regards to rule number, performance and computational time.

Finally, the evaluation of IConAS approach as a whole was performed in the Elderly Healthcare domain. This evaluation started by the ***Treatment Implementation (T4)*** to demonstrate how the IConAS approach was implemented. This evaluation was finished by the ***Implementation Evaluation (T5)*** to test the feasibility of the implementation from a functional point-of-view by considering healthcare-related scenarios and from a non-functional point of view (i.e., activity recognition performance and user satisfaction).

1.5 Thesis Outline

The present thesis is divided in four parts that group related chapters. The structure of the thesis and related chapters follow the research tasks previously described in the adopted methodology's regulative cycles in subsection 1.4.2 and the aforementioned research questions in section 1.2. Accordingly, the structure and the content of each of the chapters are as follows:

Part I describes the ***Problem Investigation (T1)*** and includes:

- ***Chapter 1*** starts the description of the motivations and the problems that justify the need of intelligent context-aware solutions in dynamic environments. It, next, introduces the different objectives and research questions. Then, it highlights the several contributions;
- ***Chapter 2*** introduces the main concepts that are involved in the development of this thesis such as pervasive computing, context-awareness and ontology engineering basics. This chapter also presents the foundations of rule learning. Finally, the chapter describes some details about Abraham Maslow's Hierarchy of Human Needs and about how they can come into making decisions;
- ***Chapter 3*** presents a systematic review of the literature of the topics studied. We start investigating works that have proposed pervasive middleware solutions. Next, we focus on analyzing ontology learning approaches that can provide relevant insights for context evolution. And then, we present a review on the rule learning approaches. In order to analyze the suitability of these works, we propose several criteria and we analyze each presented work accordingly. Finally, we compare all the proposals in order to justify that the main problem described by this thesis has not been completely solved yet;

Part II designs a solution to the main research contributions stated in section 1.3. In this part, we address the ***Treatment Design (T2)*** and the ***Treatment Validation (T3)***. Therefore, this part involves three chapters:

- ***Chapter 4*** describes IConAS approach that combines two approaches to augment an existing middleware solution in order to support dynamic environments at runtime;
- ***Chapter 5*** presents the CoE approach, which forms the first part of the IConAS approach. This approach aims to address the automatic evolution of context models at runtime according to changes occurring in the enclosing dynamic environments.
- ***Chapter 6*** describes the DMA approach, which forms the second part of the IConAS approach. This approach aims to generate new decision rules that could answer the evolving context models at runtime;

Part III describes the research activities that have been carried out to validate the solutions proposed in Part II and describes the available tool support for these solutions. This part contains:

- ***Chapter 7*** that addresses tasks ***Treatment Validation (T3)***, ***Treatment Implementation (T4)*** and ***Implementation Evaluation (T5)***. This chapter presents a series of experiments that aim to validate the proposed approaches;

Part IV closes the thesis with the conclusions, main limitations and possible future directions.

- ***Chapter 8*** discusses the conclusions drawn from the research. Next, it lists the publications derived from the research presented in this thesis. As a final point, the limitations and the main lines of future work are presented.

CHAPTER 2

Background

2.1 Introduction

The present thesis relies on different background knowledge to deal with the development of an approach to provide intelligent context-aware solutions that support automatic context evolution and decision-making adaptation in dynamic environments. In this chapter, we present key background knowledge that is required to place this thesis into a broader perspective.

This chapter is structured as follows. In section 2.2, we first give a clear definition of pervasive computing. In section 2.3, we present the context definition and we briefly review the key context modeling approaches. Section 2.4 provides an overview of ontology engineering, more in particular ontology learning and ontology evolution, as they form the foundation of context evolution and the context modeling approach chosen in this thesis. Section 2.5 shortly goes through the foundations of rule learning. In section 2.6, we focus on specific aspects of Abraham Maslow's Hierarchy of Human Needs. Finally, section 2.7 concludes the chapter with short concluding remarks.

2.2 Pervasive Computing

The first research efforts in the field of ubiquitous computing started in 1988, at Xerox Palo Alto Research Center (PARC) [Weiser et al., 1999]. They began with a novel paradigm of desktop computing that focuses on the use of personal computers with high computational power. That paradigm is still in use and it functions well for a wide range of tasks. However, the researchers from PARC identified the following shortcomings of the personal computers: "too complex and hard to use; too demanding of attention; too isolating from other people and activities; and too dominating as it colonized our desktops and our lives." [Weiser et al., 1999]. Later, they have experienced the advent of the paradigm of ubiquitous computing, that aims to address desktop computing's problems by intertwining the computing technologies with everyday life to the extent when the technologies become indistinguishable from it [Weiser, 1991]. "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." These words were written by Mark Weiser in his paper "The Computer for the 21st Century" [Weiser, 1991]. Weiser and his colleagues in PARC have used the term "ubiquitous computing" to describe their vision of an embedded computing world. Within this world, computers would be an invisible part of our everyday lives; and computing was going to change from desktop, personal computing to a more distributed, mobile and embedded form. Ubiquitous computing solutions are now becoming an integrated part of the everyday environment. Various implementations of the ubiquitous computing paradigm includes smart homes, smart offices and other ambient intelligence solutions, wearable computing devices, personal digital assistants, social networks. However, there is still much research to be done in the area. The concept of pervasive computing is connected to the concept of ubiquitous computing so closely that those terms are sometimes used interchangeably, there is a nuance between them [Poslad, 2011]. It was noted that "the vision of ubiquitous computing and ubiquitous communication is only possible if pervasive, perfectly interoperable mobile and fixed networks exist." [Weyrich, 1999]. Although often used as synonyms, the term "pervasive computing" is often preferred when discussing the integration of computing devices and weaving them into the everyday environment, while the term "ubiquitous computing" is usually preferred when addressing the interfaces and graceful interaction with the user. In computing terms, those seem like somewhat similar concepts. Based on the provided definitions, as [Weiser, 1991], we consider that "pervasive computing" and "ubiquitous computing" refer to the same thing. So, this thesis is mostly focused on pervasive computing challenges, and therefore the term "pervasive computing" is used in most cases. Figure 2.1 shows the pervasive computing environment with its main features. As shown in Figure 2.1, pervasive computing environment consists of devices, middleware and solutions, such as applications, systems or frameworks. Furthermore, pervasive computing has evolved into a more general paradigm known as context-awareness. Detailed information about context-awareness is given in the following subsection.

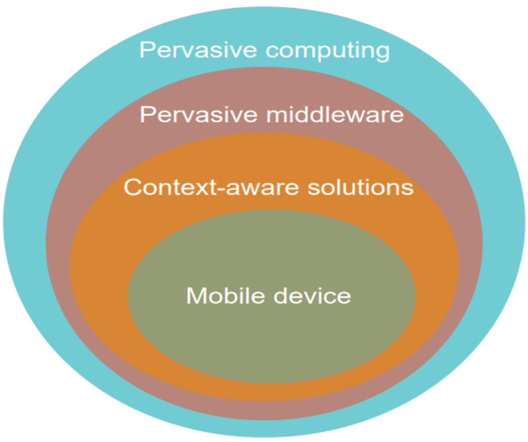

Figure 2.1: Pervasive computing environment.

2.3 Context-Awareness

2.3.1 Context Definition

Within the pervasive computing domain, the notion of context becomes increasingly important. User needs have to be supported actively at runtime and context plays an increased role in supporting user needs efficiently [Brézillon, 2003]. There are several various definitions of context that have been put forward. The generic and reasonable definition given by Dey and Abowd [Abowd et al., 1999] is nowadays used as reference in most literature on context awareness: "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 application themselves." Dey and Abowd also give four primary context categories, but are not limited to:

- Identity (who's);
- Location (where's);
- Time (when's);
- Activity (what's).

Due to the broad and general Dey and Abowd's definition that focuses on the user, it might be difficult to understand what types of context information should be designed considering a mobile user's context in a dynamic environment. Therefore, a more detailed definition of context is needed in order to better understand the context of mobile users. In the domain of mobile applications, where modern mobile devices have built-in sensors (e.g., GPS, accelerometer), use external sensors (e.g., heart rate sensor) connected to the mobile device via Bluetooth, utilize smartwatch sensors, activity tracker bands, and have access to the Internet, we need to consider the context related to environment in this thesis. An example of generic categorizations considered in this thesis:

- ***User context-*** refers to any kind of context information related to the user. User context information can be the user's preferences, age, location, medical history, etc;
- ***Physical context-*** consists of any kind of context information related to the physical environment, in opposition to the computing environment. Physical context information is lighting, temperature, noise level, weather and so on;
- ***Temporal context-*** defines any kind of context information related to time. Time of day, date and season are typical temporal context information.

2.3.2 Context Modeling Approaches

For context aware systems, context information has to be represented in a model. On the formal basis of a context model, standardized access and communication is possible. The context model therefore specifies the data structures and extension possibilities for systems. Therefore, [Chen and Kotz, 2000] identified six context modeling approaches that differ in their structure and presentation of the context information. The following subsections give an overview over the six approaches.

2.3.2.1 Key-Value

Key-Value models are the simplest modeling approach [Moore et al., 2007, Tzanavari et al., 2008, Wei and Jin, 2012] and that is why it has some popularity [Bellavista et al., 2012]. In this approach, each piece of context information is assigned as a list of attributes [Villegas and Müller, 2010]. A key is then the attribute name and the value represents the value corresponding to this attribute [Villegas and Müller, 2010, Bellavista et al., 2012]. While the value may change, the key always stays the same. Different formats for Key-Value pairs are possible, such as text files or binary files [Surve and Ghorpade, 2017]. With its simple nature, a Key-Value model is very easy to implement and to manage [Esseynew et al., 2016]. Furthermore, it is useful for building prototypes and running on devices with low computing power and storage [Hodaie et al., 2013]. While the simplicity makes the implementation very easy, there are also disadvantages that make this approach less attractive for context modeling. The Key-Value modeling approach is incapable of modeling sophisticated, complex information [Wei and Jin, 2012] and to retrieve efficient context [Esseynew et al., 2016]. This approach also lacks capabilities for structuring context information and mechanisms to check its validity [Bellavista et al., 2012]. And according to [Surve and Ghorpade, 2017], the Key-Value model does not scale well.

2.3.2.2 Markup Scheme Models

In Markup Scheme Models, a hierarchical data structure is used, which consists of tags, attributes and content (i.e., values) [Tzanavari et al., 2008]. Markup Scheme Models are based on using markup languages to represent context information. The most well-known markup languages are the Hypertext Markup Language (HTML) and the Extensible Markup Language (XML) [Moore et al., 2007], which fall, among others, under the umbrella of the Standard Markup Language (SGML) that is the super class of all the markup languages [Tzanavari et al., 2008]. Markup based modeling is more structured than the simple key-value based approach, as the used markup language is usually accompanied by a schema. This allows for certain kinds of validation, such as type and range checks for numeric values and leads to better context retrieval capabilities. The advantages of Markup Scheme Models are that they can handle heterogeneity and incompleteness [Esseynew et al., 2016], context information can be validated by means of validation tools such as XML-schemas and data can be structured via nested XML structures [Bellavista et al., 2012] allowing the expression of complex relations. Furthermore, Markup Scheme Models allow for an efficient data retrieval [Surve and Ghorpade, 2017]. The main disadvantage of Markup Scheme Models' is that they do not allow reasoning. There is a lack of design specifications, which means that context modeling, data retrieval, interoperability and reusability over different markup schemes can be difficult [Surve and Ghorpade, 2017]. Furthermore, Markup Scheme Models lack expressive structure, have weak formalism and are inadequate for capturing context information and relationships [Esseynew et al., 2016].

2.3.2.3 Graphical Models

Graphical models are used to model context with relationships [Surve and Ghorpade, 2017]. Contextual information is represented using graph data structures and richer data types [Esseynew et al., 2016]. A very well-known graphical modeling instrument is the Unified Modeling Language (UML). Due to its generic structure, UML is appropriate to model context information. The advantages of Graphical models are that they are more expressive than Key-Value and Markup Scheme Models, as relationships are captured into the model [Esseynew et al., 2016]. An important advance over markup-based models is the ability to explicitly indicate relationships that hold between context information. In addition, they have a good balance between expressive power and efficient reasoning and a good support for software engineering. Furthermore, they are easy to learn and use [Wei and Jin, 2012]. However, the disadvantages of Graphical models are that their lack of support for hierarchical context description [Wei and Jin, 2012], lack of model semantics leading to limited reasoning and generalization support, and lack of support for distributed context modeling [Esseynew et al., 2016].

2.3.2.4 Object-Oriented Models

Object-Oriented models aim to retain the basic paradigms of object orientation: encapsulation, reusability and inheritance to cover parts of the problems arising from the dynamics of the context in pervasive environments. Structurally related context information is encapsulated in an object's internal state and can only be accessed by its public interface. The model can easily be extended by new information types (classes) or updated information (object). The most important advantage of Object-Oriented models is that the interactions between context data and the services (e.g., context-aware systems) are easy, because the same abstractions as in object-oriented programming languages are used and this simplifies the implementation of context [Bellavista et al., 2012]. Also, low-level context information is processed in a satisfying way to be able to infer high-level information from it [Hodaie et al., 2013]. However, high computation power is required to be able to handle the complexity of the object-Oriented context models [Esseynew et al., 2016]. Furthermore, the Object-Oriented model approach does not support built-in reasoning capabilities and lacks explicit schema or semantics of the model.

2.3.2.5 Logic-Based Models

Logic-based modeling is the most formal approach to context modeling. In a Logic-Based model, the context is consequently defined as facts, expressions and rules. Usually contextual information is added to, updated in and deleted from a logic-based system in terms of facts or inferred from the rules in the system respectively. In other words, changes in the environment are reflected in the model by adding, removing or updating the facts and respective rules. Logic-Based models have a high level of formality [Esseynew et al., 2016], this is their biggest advantage. They also provide a lot of expressiveness compared to the other models [Esseynew et al., 2016]. However, they do not offer simple functionalities to deal with data validity [Bellavista et al., 2012]. Moreover, reasoning is possible up to a certain level [Surve and Ghorpade, 2017]. In Logic-Based models, heterogeneity and incompleteness are still lacking [Esseynew et al., 2016]. Furthermore, [Hodaie et al., 2013] state that Logic-Based models are more heavyweight than, for example, Key-Value or Markup models.

2.3.2.6 Ontology Models

The previously mentioned modeling approaches can, according to [Moore et al., 2007], be seen as precursors to ontology modeling, since ontology makes use of many of the properties that are used in the other modeling approaches. The ontology modeling approach is a representation way to specify information as concepts and their relationships [Wei and Jin, 2012] with the usage of semantics [Surve and Ghorpade, 2017]. As the context may be considered as a specific kind of knowledge, it can be modeled as ontology by representing the contextual information in a machine-readable form in a data structure [Moore et al., 2007]. Thus, ontology models are very appropriate for mapping every-day knowledge and expressing complex relationships "within a data structure easily usable and manageable automatically" [Bellavista et al., 2012]. This is due to the fact that they have a high formal expressiveness and specification of context parameters and relations between context features in a particular domain [Villegas and Müller, 2010] and the possibilities for applying ontology reasoning techniques [Bellavista et al., 2012]. Examples of possible data structures are Resource Description Framework (RDF), Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL) [Surve and Ghorpade, 2017]. With the rising interest in ontological modeling and the availability of OWL as description language, a number of tools have been developed to support the work with ontologies. Protégé [Musen et al., 2006] is a graphical editor and Jena [McBride, 2002] provides an API to access OWL ontologies in Java. Furthermore, ontologies make use of powerful reasoning techniques [Tzanavari et al., 2008] and they are able to handle heterogeneity [Esseynew et al., 2016].

2.4 Basics of Ontology Engineering

Ontology engineering is defined as a collection of activities that focus on methodologies for developing and maintaining ontologies [Gómez-Pérez et al., 2006]. It involves several activities as noted by [Suárez-Figueroa et al., 2012]. In this section, we concentrate on describing the major activities, such as ontology learning and ontology evolution, as these provide a solid basis for investigating the research question ***RQ.1.***.

2.4.1 Ontology Learning

Ontology learning is a process of developing a conceptualization of a domain with less human intervention and more automatic or semi-automatic approach, i.e., generating ontology from a scratch, enriching or populating an existing ontology. Ontology enrichment is the task of extending an existing ontology with additional concepts and semantic relations and placing them at the correct position in the ontology, whereas ontology population is the task of adding new instances of concepts to the ontology [Petasis et al., 2011]. Ontology learning also means automating a process by which discovering, creating, extracting of ontological knowledge. The input for ontology learning can be distinguished by completely unstructured, semi-structured or structured data. The output of the ontology learning is grouped together as “concepts, taxonomic relations, non-taxonomic relations, individuals, axioms” [Cimiano, 2006]. In this sense, we have focused our interest in this field, which could be applied in Chapter 5 and might be useful for context evolution.

2.4.2 Ontology Evolution

Ontology evolution has been defined in various ways. [Haase and Stojanovic, 2005] see ontology evolution as the process to ‘adapt and change the ontology in a timely and consistent manner’. Otherwise, [Flouris et al., 2008] define ontology evolution as a process aiming to ‘respond to a change in the domain or its conceptualization’ by implementing a set of change operators over an existing ontology. Apart from that, the NeOn methodology for ontology engineering states that ontology evolution is 'the activity of facilitating the modification of an ontology by preserving its consistency'. A common characteristic of the above definitions is that they have a strict view on ontology evolution focusing only on updating an existing ontology based on the required changes and therefore they see ontology learning as a separate activity. We argue that ontology learning is intrinsically linked to the ontology evolution task and therefore should be considered when discussing ontology evolution. As a result, in this thesis, we adopt a broader view of ontology evolution encompassing the changes made to an ontology and ontology learning.

2.5 Foundations of Rule Mining

2.5.1 Data Mining

The concept of data mining refers to the process of analyzing large data sources in order to extract hidden knowledge and implicit interesting patterns. Data mining outcomes or models are used in different operations such as decision making, status analysis and prediction of behaviors. Yet, the term "Data Mining" was not introduced until the early 1990s. The roots of data mining can be traced back to three scientific fields: statistical studies, artificial intelligence and machine learning. The vision and understanding on how to extract useful information from data evolved as these different fields evolved. There are basically two most important reasons that data mining has attracted a great deal of attention in recent years. First, the capability to collect and store the huge amount of data is rapidly increasing day by day. The second reason is the need to turn such data into useful information and knowledge. The knowledge that is acquired through the help of data mining can be applied into various applications like business management, retail and market analysis, engineering design and scientific exploration [Ogunde et al., 2011]. Technically, data mining is the process of finding correlations or patterns among many fields in large data. Data mining techniques have comprehensive analytics and powerful approaches that allow tackling and analysis of more complex data.

2.5.2 Association Rule Mining

Association rules mining is an important and well-researched problem in data mining that has shown great potential and captured attention since it was first introduced in the early 1990s [Kotsiantis et al., 2006]. Its importance rises from its capabilities to discover unapparent relations, interesting correlations, frequent patterns, associations or casual structures among data in large data sources. More specifically, it aims to find out hidden relationships among items in data sources. [Agrawal et al., 1993] introduced association rule mining as an unsupervised approach. They underline the idea for generating association rules from large scale transaction data in the context of the market basket problem. Association rules mining algorithms, such as Apriori and FP-Growth, have proven to be quite useful in many diverse fields including Web usage mining, intrusion detection, bioinformatics, etc. Both algorithms are described in the following two subsections as they have great practical use and will provide a solid basis for comparison.

2.5.2.1 Apriori Algorithm

Apriori algorithm is a well-known algorithm for finding frequent item sets from a set of data by using candidate generation. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level-wise search, where k-item sets are used to explore (k+1)-item sets. First, the set of frequent 1-item sets is found by scanning the data source to accumulate the count for each item and collecting those items that satisfy minimum support. The resulting set is denoted L1. Next, L1 is used to find L2, the set of frequent item sets, which is used to find L3, and so on, until no more frequent k-item sets can be found. The finding of each Lk requires one full scan of the data source [Parekh et al., 2015].

2.5.2.2 FP-Growth Algorithm

Frequent Pattern Growth, or simply FP-Growth algorithm, adopts a divide-and conquer strategy as follows. First, it compresses the data source representing frequent items into a frequent pattern tree, or FP-tree, which retains the itemset association information. Second, it divides the compressed data source into a set of conditional data sources. each conditional data source is associated with one frequent item or "pattern fragment". Then, it mines each data source separately. For each "pattern fragment" only its associated data sources need to be examined. Therefore, this algorithm may substantially reduce the size of the data sources to be searched, along with the "growth" of patterns being examined [Liu, 2011].

2.5.2.3 Association Rule

Association rules provide information in the form of "if-then" statements. This rule "if X then Y" is computed from data sources and, unlike the if-then rules of logic, association rules are probabilistic in nature. The left-hand side of the rule X (the "if" part) is called the antecedent of the rule and the right-hand side of the rule Y (the "then" part) is called the consequent of the rule.

2.6 Abraham Maslow's Hierarchy of Human Needs for Decision-Making

Support for middleware solutions and context-aware solutions in their decision-making processes requires some theories to assist them to make better decisions. In this thesis, this support can be provided by offering a broad view of Abraham Maslow's Hierarchy of Needs theory [Maslow, 1998] that can be applied for determining decision-making. Maslow's Hierarchy of Human Needs is a theory established by an American Psychologist named Abraham Harold Maslow. This theory is a basic needs model that shows needs having to be met. These basic needs, however, are weak needs, quiet biological urges that are often confused and easily overlooked in day-to-day affairs. It means that individuals seek to satisfy the higher needs that occupy a set of hierarchy. [Tamang, 2015] argue that Maslow's hierarchy provides a solid footing to understand the needs faced by people seeing that each Maslow's level of need has relevance for people assistive technologies [Thielke et al., 2012]. The hierarchy of needs proposed by Maslow categorizes human needs into 5 distinct levels:

- ***Physiological needs.*** According to Maslow, the physiological needs are the first needs that have to be fulfilled. They are the obvious needs such as food, water, air, sleep, etc. Every human requires all needs above because they are the basic needs of a human being. Therefore, satisfying those need is so important for human to survive [Duanel, 1977],
- ***Safety needs.*** When all physiological needs are satisfied and are no longer controlling thoughts and behaviors, the security needs can become active [Boeree, 2006]. These needs represent a need for safety or security in our environment. These needs include security, stability, and freedom for fear and anxiety [Duanel, 1977]. Maslow believed that, like physiological needs, safety needs are primarily triggered in emergencies,
- ***Love and belonging needs.*** If both the physiological and the safety needs are fairly well gratified, then there will emerge the love and belongingness needs [Maslow, 1981]. These needs can be expressed through a close relationship with a friend, lover, mate or through social relationship formed within a group [Schultz and Schultz, 2016],
- ***Esteem needs.*** If the love needs have been adequately met, the esteem needs become dominant. People require esteem and respect from ourselves, in the form of feelings of self-worth, and from other people, in the form of status, recognition or social success,
- ***Self-actualization needs.*** When one has satisfied the first four-level of need, the final level of development, which Maslow termed self-actualization, can be reached. To satisfy the self-actualization needs, people must be free of constraints imposed by society and by ourselves, not be distracted by the lower order needs, be secure in our self-image and in our relationships with other people, be able to love and be loved in return and have a realistic knowledge of our strengths and weaknesses [Schultz and Schultz, 2016].

2.7 Concluding Remarks

In providing a context for understanding the thesis topic, this chapter outlined the different essential theoretical foundations, such as pervasive computing, context-awareness, ontology engineering basics and so on, in which the research is based. These different theoretical foundations, presented in this chapter, lays the foundation for the thesis' contributions. These theoretical foundations were used to design IConAS approach in order to offer intelligent context-aware solutions that reflect changes in dynamic environments at runtime, which is described in chapter 4.

This chapter presented the following:

- Firstly, in section 2.2, we have introduced the concept of pervasive computing.
- Secondly, in section 2.3, we have provided an overview of the notion of context, the different context modeling approaches that are used to present context information.
- Thirdly, in section 2.4, we have given an overview of different activities that are used to develop and evolve ontologies. In chapter 5, we will present our contribution to this topic, where we will exploit a context evolution based on ontology.

- Fourthly, in section 2.5, we have presented an introduction to data mining and specifically the association rule mining that are of particular importance for chapter 6. In addition, we have discussed two important association rule mining algorithms.
- Finally, in section 2.6, we have briefly discussed Maslow's Hierarchy of Needs and its main contributions. This hierarchy is used in chapter 4, where a case study for elderly activity recognition and healthcare decision-making is introduced and the service recommendation based on Maslow's need levels is described.

CHAPTER 3

State of the Art

3.1 Introduction

According to the design science guidelines in [Wieringa, 2009, Wieringa, 2014] and our Thesis's DSRM in Figure 1.3, the second step of the engineering cycle is to design a solution to a practical problem. However, it may occur that the researcher does not have to invent something new but make an inventory of solutions available in the literature. Alternatively, the researcher can assemble a solution from known solutions. This is why state-of-the-art studies are mandatory to ascertain that there exist no previous solutions to the problem at hand before proposing a new one. Based on the different research objectives and questions described in this thesis, this state-of-the-art focuses on the discussion of three lines of research on which this thesis relies.

In section 3.2, we first review the literature to find out whether there is a pervasive middleware solution available for use that could cope with runtime changes occurring in the dynamic environments. The intention of this section is to explain why existing middleware solutions fail to serve as a solid solution for fully automatic context evolution and decision-making adaptation at runtime. Since no middleware solution was available, we will opt to provide IConAS approach that augments an existing solution with these capabilities at runtime to offer intelligent context-aware solutions. With regard to the findings in section 3.2, section 3.3 focuses on the study of the state-of-the-art in context evolution topic. It provides an overview of approaches in ontology learning, their properties and their details. Then, section 3.4 describes the state-of-the-art in relation to the rule learning approaches that have made use of data for generating new rules. Finally, section 3.5 closes this chapter with some concluding remarks.

3.2 Pervasive Middleware

Last two decades have been the peak period in the development of pervasive middleware [Chen et al., 2012] since pervasive middleware is essential for two reasons according to [Bandyopadhyay et al., 2011]:

1. Act as a relationship to merge different heterogeneous components together.

2. Hide the complexity that can be emerged in different context-aware applications.

Over the years, there have been several surveys carried out in relation to the pervasive middleware field. The availability of extensive surveys represents an indicator for the seniority of pervasive computing research [Abowd, 2012, Dourish and Mainwaring, 2012]. At the same time, the continuous emergence of these surveys shows that the search for a suitable middleware, capable of offering intelligent context-aware solutions, is still ongoing. In light of the great potential and continuous advances, we survey numerous middleware specifically developed for pervasive computing mainly for three reasons:

- to compare the existing middleware solutions;
- to identify shortcomings of the existing middleware solutions in relation to the identified objectives previously stated in section 1.2;
- to show the relevance of the problems handled in this thesis for the research community.

The present section is structured into two parts. The first part surveys relevant middleware from several surveys [Baldauf et al., 2007, Perera et al., 2013, Madhusudanan et al., 2018, Pradeep and Krishnamoorthy, 2019] (see subsection 3.2.1). The second part provides a discussion on the surveyed middleware solutions (see subsection 3.2.2).

3.2.1 Middleware Survey

Context-Toolkit
An important middleware architecture for building context-aware applications was presented by Dey et al. [Salber et al., 1999]. Context-Toolkit enables rapid prototyping of context-aware applications that implement pervasive computing scenarios. It is one of the first middleware that identifies the advantage of separating sensor context acquisition from the application logic for facilitating the creation of orchestration services and for enabling portability via exchanging information only via context and offering a uniform interface to access the context. The architecture of Context-Toolkit consists of sensors to collect context information, widgets, which act as repositories for their context, to encapsulate the contextual information and provide methods to access the information, as well as interpreters to transform the context information into high-level formats that are easier to handle. For this purpose, key-value modeling is used for context modeling. Moreover, Context-Toolkit provides only basic functionality for context access. It does not provide extensibility of the core functionality via services.

GAIA
Gaia [Román et al., 2002], another middleware infrastructure, extends typical operating system concepts to include context-awareness. It provides a framework to develop user-centric, resource-aware, multi-device, context-sensitive and mobile applications to support context-aware agents in smart spaces. The three major building blocks of Gaia: Gaia Kernel, Gaia Application Framework and Applications. Gaia Kernel block supports various forms of context-awareness and includes:

- **A context service.**, which allows applications to find providers for the context information they require.
- **An event manager service**, which monitors the entities, such as people, hardware and software components, entering and leaving a smart space.
- **A space repository service**, which maintains descriptions of hardware and software components.
- **A context file system**, which associates files with relevant context information and dynamically constructs virtual directory hierarchies according to the current context.

CARISMA

Context-aware Reflective Middleware System for Mobile Applications (CARISMA) is provided by Capra [Capra et al., 2003]. CARISMA is proposed as a mobile computing middleware that enhances the construction of adaptive and context-aware applications. The middleware enables the developer to describe context information in key-value pairs by means of XML, which is defined as a policy. As a result, the application is allowed to dynamically inspect the middleware behavior and dynamically change its behavior by means of a meta-interface that enables runtime modification of the internal representation that was previously made explicit. CARISMA exploits the use of computational reflection to achieve dynamic adaptation to context changes. The CARISMA middleware is responsible for maintaining a valid representation of the execution context by directly interacting with the underlying network operating system. CARISMA uses aspects weaving of functional concerns, which does not suit the context-driven behavior. As described before, this requires context-driven aspects supported by context handling aspects in the platform. In addition, it supports parameter-based adaptation using an internal adaptation approach. Furthermore, CARISMA is specific to applications in which context changes are foreseen and planned for their anticipation at the design time of the software.

MobiPADS

MobiPADS [Chen et al., 2004a] is designed to support context-aware processing by providing an executing platform to enable active service deployment and the reconfiguration of service compositions in response to varying contexts in the operating environment. It supports dynamic adaptation at both the middleware and application layers to provide a flexible configuration of resources to optimize the operations of the mobile applications. Within the MobiPADS system, a series of mobilets is linked together to form a processing chain called the service chain, which reacts and adapts to the varying characteristics of a wireless environment. In the MobiPADS service space, mobilets exist in pairs: a master mobilet resides at the MobiPADS client and a slave mobilet resides at the MobiPADS server. Mobilets access the services of the system components through the mobilet API, which also provides interfaces to allow the system components to communicate and configure the mobilets. At the top level of the service space, there is a set of meta-objects that reflects the configuration for the composite events and service chain, as well as the adaptation policies. The MobiPADS achieves context-awareness by using an event notification model, which monitors the status of all contexts of interest and reports the event to the subscribed entities. These include all of the entities within the platform such as the system components, the mobilets, and the mobile application. On detecting changes in the environment, the MobiPADS system can respond either by reconfiguring the current service chain or by communicating the changes to each of the mobilets. An abstraction of service object interactions and configurations is expressed in a high-level declarative language written in XML format. Based on these profiles, the MobiPADS system can respond to changes of context by adding and removing mobilets within the service chain to select an optimum set of mobilets. Simply adding or removing mobilets within the service chain may not be enough to adapt to contextual changes. To allow a finer-grained adaptation, the MobiPADS system allows the mobilets to subscribe to an event and react to the event message by adjusting its internal parameters to best adapt to the changes.

CoBrA

A Context Broker Architecture (CoBrA) [Chen et al., 2003] discusses architecture for supporting context-aware systems that provide pervasive computing services to users and may include physical spaces like living rooms, vehicles, classrooms and meeting rooms. The architecture of CoBrA is aimed to support context-aware systems in smart spaces. CoBrA uses OWL to define ontologies for context modeling and reasoning. A rule-based method is applied for context interpreting and reasoning. The discussed architecture has a core server entity called Context Broker that maintains and manages a shared contextual model. The context broker consists of four functional main components:

- **Context Knowledge Base** that provides a centralized model of context.
- **Context Inference Engine** that reasons about contextual information that cannot be directly acquired from the sensors.
- **Context Acquisition Module** that acquires contextual information.
- **Privacy Management Module** that detects and resolves inconsistent knowledge that is stored in the shared model of context and protects user privacy.

SOCAM

SOCAM [Gu et al., 2004] is a framework of a Service-Oriented Context-Aware Middleware to build context-aware mobile services in pervasive computing environments. It is a distributed middleware that converts various physical spaces from which contexts are acquired into a semantic space where contexts can be shared and accessed by context-aware services. It mainly uses ontologies to model the context. Consequently, SOCAM supports semantic representation, context reasoning and context-knowledge sharing. It consists of the following components:

- **Context providers.** They abstract useful contexts from external or internal sources and convert them to OWL representations so that contexts can be shared and reused by other service components.
- **Context interpreter.** It provides logic-reasoning services to process context information.
- **Context database.** It stores context ontology, which is divided into two parts. The upper ontology that contains general concepts, and different domain-specific lower-level ontologies.

- **Context aware services.** They make use of different levels of contexts and adapt the way they behave according to the current context using the inference engine provided by the ontology.
- **Service-locating service.** It provides a mechanism where context providers and the context interpreter can advertise their presence. It also enables users or applications to locate these services.

The main contribution of the SOCAM is classifying pervasive computing into different context ontology domains such as vehicle or home. Each SOCAM domain has specific context models. The use of ontologies allows extending the context meta model with additional semantics with the help of experts at run time.

MADAM

Mobility and ADaptation enAbling Middleware (MADAM) was created to build adaptive applications for mobile devices using architecture models [Mikalsen et al., 2006]. MADAM enables to build an adaptive application that is able to be reconfigured automatically at runtime to adapt itself to context changes in a Pervasive mobile environment. This approach describes three main services. The first service is the Context manager which monitors user requirements and context's changes for detecting relevant changes and notifying them to the Adaptation manager. The second service is the Adaptation manager which is responsible for reasoning about context's changes, for determining when there is a need to launch the adaptation process and for selecting the best application variant that corresponds to the current context. And the last service is the Configurator which is responsible for adapting applications by reconfiguring the architectural model. The architecture model presented at runtime allows generic middleware components to reason about and control the adaptation strategy. The MADAM middleware can support planning-based adaptation by instantiating a plugin architecture that fulfills the utility function evaluations.

MUSIC

Mobile USers In Ubiquitous Computing (MUSIC) middleware [Rouvoy et al., 2008] is an extension of the MADAM component-based planning framework that optimizes the overall utility of applications when context-related conditions occur. MUSIC middleware supports various adaptation mechanisms, such as setting configuration and application parameters, replacing components and service bindings and also redeploying components of the distributed computing infrastructure. It is an extension of the earlier project MADAM, and inspires its concepts from the main ones of MADAM. In addition to what MADAM has achieved, MUSIC also models and realizes service bindings and extended situational contexts as part of the context dependencies based on a service-oriented approach. This means that if some appropriate service is detected at runtime in the execution environment, it can automatically be integrated and can replace another software component. Moreover, MUSIC tackles pervasive computing environments and Service Oriented Architectures (SOA). The decision-making process in MUSIC is similar to MADAM middleware, both use a utility function to evaluate all the reasoning dimensions used by the adaptation reasoner to select and deploy the component implementation, thereby providing the best utility. The process of adapting applications in response to changes of context includes a planning procedure and a reconfiguration process.

CAMPUS
[Wei and Chan, 2013] proposed an approach for automated context-aware adaptation decisions, at runtime, by a middleware layer. The resulting Context-Aware Middleware for Pervasive and Ubiquitous Service (CAMPUS) operates according to runtime contextual information. This middleware advocates automated run-time adaptation decisions with the matching of three technologies: compositional adaptation, ontology and description logic/first-order logic reasoning. It has taken an enormous step to advocate automated run-time adaptation decisions instead of depending on predefined adaptation policies that only take limited contextual changes potentially operating in a dynamic situation. CAMPUS, as a typical layered architecture, consists of three tiers: the programming layer, the knowledge layer, and the decision layer.

- **Programming layer.** It is responsible for constructing and reconfiguring context aware applications by adopting the instructions from the decision layer.

- **Knowledge layer.** Three ontologies including Context Model, Tasklet Model, and Service Model are proposed to represent semantics of knowledge which is necessarily required by CAMPUS to make adaptation decisions. The knowledge could be the requirements desired by target service, the properties of the available tasklets, the context requirements imposed by tasklets, and the properties of run-time context.

- **Decision layer.** Decision makers use a multi-stage normative decision model, which includes preprocessing, screening and choice, to choose the best tasklet alternatives for a given task. The automated adaptation decisions will be forwarded to the programming layer.

CAMPUS provides an effective middleware solution for integrating context awareness to application development. Thus, it could automatically derive context-aware adaptation decisions at run time by means of semantic-enhanced decision making.

SeCoMan
SeCoMan [Celdrán et al., 2014], as the abbreviation of Semantic Web-based Context Management, is a context-aware framework for developing smart applications where users can share their location to the right users, at the right granularity, at the right place, and at the right time. It uses semantic Web for data description, data modeling and reasoning of things in the IoT context. To develop context-aware smart applications, SeCoMan architecture, which is used to offer predefined queries and context-aware services, is presented. A layered architecture is used in this framework. The three main layers of SeCoMan framework are:

- **Application layer**, where different applications reside on top of SeCoMan in order to offer desired services for users.

- **Context management layer**, which is responsible for context-aware location-related services. The context awareness of this framework is to provide the location of the users, and by using semantic rules, it maintains the privacy of the user's location with restriction to its access. The reasoner module receives ontological models generated by the Interpreter module and returns inferred models with new knowledge by a set of predefined rules.

- **Plug-in layer**, which is responsible for providing extensibility to SeCoMan. This layer is composed of different plug-ins that interact with the Middleware module, which communicate with sensors or other devices to receive context information, and with the Location Systems module to obtain information about the environment.

Context-aware knowledge based middleware
A context-aware information management middleware [Evchina et al., 2015] that addresses selective information delivery with respect to the user's role in the system. An ontology-based technique for context modeling and organization was used in order to describe the environment. In addition, a semantic reasoner was applied in context reasoning.

CONASYS
Context-Aware System called CONASYS [de Matos et al., 2017] that enables environment sensing and provides relevant contextualized services to the users in an IoT domain. It provides an API to connect to users. Key–value and Markup scheme approaches were used to model and organize the context. Drools engine was used for rule-based reasoning. CONASYS has architecture consist of three layers:

- The first layer is the communication layer in charge for receiving and interpreting the request of users with sending results to him.
- The second layer is the storage layer responsible for storing content of all modules present in CONASYS.
- The third is the processing layer for reasoning a context.

Cooperative middleware
A cooperative middleware [Hoque et al., 2017] was proposed to provide service in time for users by inferring the current context and distributing it in a cooperative manner. Its goal is to reduce the context computation cost of providing services on time. An OWL-based ontology technique was used to define the context semantically. Moreover, a combination of rule-based and ontology-based were used to reason the context. There are three main layers in hierarchical design which are context provider layer, middleware layer and application layer.

- **Context provider layer** performs the task of context acquisition, mapping and representation. It produces low-level context entity from sensing raw data using mapping technique.
- **Application layer** generates smart home services using context with IF-THEN condition.
- **Middleware layer** behaves like a gateway to hide interaction complexities.

AUM-IoT
Adaptive Ubiquitous Middleware-driven context-aware IoT ecosystem, AUM-IoT [Pradeep et al., 2021] primarily considers healthcare and transportation domains and provides relevant context-aware services by utilizing the functionalities and capabilities of fog and cloud computing via multi-communication protocol bridge. The AUM-IoT architecture consists of five layers:

perception, middleware, fog, cloud and application. First, the perception layer is responsible for perceiving information in raw form from actuators, sensors, data sources, devices and so forth. Second, the middleware and fog layers deal with context processing. Third, the cloud layer performs high computation tasks and permanent storage of context information to produce useful services. Fourth, the application layer comprises the client devices and the applications associated with the ecosystem.

3.2.2 Middleware Comparative Study and Discussion

After surveying the chosen set of pervasive middleware, it is valuable to analyze and evaluate each of them. It bears noting that the selection of the pervasive middleware that is surveyed and assessed in this section is based on the surveys [Baldauf et al., 2007, Perera et al., 2013, Madhusudanan et al., 2018, Pradeep and Krishnamoorthy, 2019]. However, these surveys do not cover all objectives that are identified in section 1.2. Therefore, in this subsection, we combine assessment criteria from the different surveys and include some additional criteria that accurately reflect the thesis's objectives. The fact that these additional criteria are not in the focus of existing surveys shows that these objectives are not well covered by pervasive middleware so far as described before in section 1.1. All assessment criteria, which represent essential requirements of a middleware, are structured into the following three levels, namely middleware design and support, context management and decision-making level. These criteria are listed and explained as follows:

- Criteria related to middleware design and support level:
  - **Context-aware.** A pervasive middleware should be aware of the users' context to provide relevant decisions and services;
  - **Adaptive.** A pervasive middleware should dynamically adapt itself at runtime;
  - **autonomous.** A pervasive middleware should enable interaction and changes even without human interference;
  - **Scalable.** A pervasive middleware should support introduction of new devices, context features, decisions and services as per the changes happening at runtime;
  - **Real-time.** A pervasive middleware should ensure timeliness by offering decisions and services in real-time.
- Criteria related to context management level:
  - **Context modeling.** A context model specifies how context information is organized, represented and stored;
  - **Context autonomy.** A context model can deal with seen and unseen changes occurring in the surrounding environments at runtime without human intervention;
  - **Context evolution.** A context model can be expandable in terms of schema and information to harmonize with changes occurring in surrounding environments at runtime. Therefore, new context features and context types can be introduced during runtime.

- Criteria related to decision-making level:
  - **Reasoning.** A reasoning can be defined as a method of deducing new knowledge and understanding better based on the available context information;
  - **Decision-making process.** A decision-making process can be either static or dynamic at runtime. Static decision-making process offers a complete knowledge base prior to runtime whereas dynamic decision-making is a process where a knowledge base is not seen as fixed but can be enriched by incoming changes at runtime;
  - **Decision-making adaptation.** The core concept behind decision-making adaptation is the general ability to extend existing knowledge bases. Automatic adaptation enables a middleware to extend existing knowledge bases by itself at runtime. In contrast, an ordinary adaptation in turn is decided and triggered by users or administrators.

After the description of each criterion, Table 3.1, Table 3.2 and Table 3.3 summarize the comparative results of the previously surveyed middleware solutions with regard to these criteria. The list of pervasive middleware is ordered based on the chronological sequence. To specify the support, the partial support, the absence and the no mention of criterion, we use $\surd$, $(\surd)$, $\times$ and - respectively.

	Context-aware	Adaptive	Autonomous	Scalable	Real-time
Context-Toolkit	$\surd$	$\times$	$\times$	$\times$	$\times$
Gaia	$\surd$	$\surd$	$\times$	$\times$	$\times$
CARISMA	$\surd$	$\surd$	$\times$	$\surd$	$\times$
MobiPADS	$(\surd)$	$(\surd)$	$\times$	$\times$	$\surd$
CoBrA	$\surd$	$\times$	$(\surd)$	$\times$	$\times$
SOCAM	$\surd$	$\surd$	$\times$	$\times$	$\times$
MADAM	$\times$	$(\surd)$	$\times$	$\times$	$\times$
MUSIC	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
Campus	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
SeCoMan	$\surd$	$\times$	$\times$	$\surd$	$\times$
Context-aware knowledge-based middleware	$\surd$	$\times$	$\times$	$\surd$	$\times$
CONASYS	$\surd$	$\times$	$\times$	$\times$	$\surd$
Cooperative middleware	$\surd$	$\surd$	$\surd$	$\times$	$\times$
AUM-IoT	$\surd$	$\surd$	$\surd$	$\times$	$\surd$

Table 3.1: Pervasive middleware comparison: Middleware design and support level

	Context modeling	Context autonomy	Context evolution
Context-Toolkit	Key-Value	×	√
Gaia	Markup schema	×	×
CARISMA	Markup schema	×	×
MobiPADS	Graphical model	×	×
CoBrA	Ontology	×	×
SOCAM	Ontology	×	√
MADAM	×	×	×
MUSIC	Ontology	√	×
Campus	Ontology	√	×
SeCoMan	Ontology	×	×
Context-aware knowledge-based middleware	Ontology	×	×
CONASYS	Key-Value, Markup schema	×	×
Cooperative middleware	Ontology	×	×
AUM-IoT	Ontology	×	×

Table 3.2: Pervasive middleware comparison: Context management level

	Reasoning	Decision-making process	Decision-making adaptation
Context-Toolkit	• Rule-based	Static	×
Gaia	• Probabilistic reasoning • Fuzzy logic	Dynamic	-
CARISMA	• Rule-based	Static	×
MobiPADS	• Events • Mobilets	Dynamic	Ordinary
CoBrA	• Rule-based • Ontology	Dynamic	Ordinary
SOCAM	• Rule-based • Ontology	Dynamic	Ordinary
MADAM	-	Static	×
MUSIC	-	Dynamic	Ordinary
Campus	• Ontology	Dynamic	Ordinary
SeCoMan	• Rule-based • Ontology	Static	×
Context-aware knowledge-based middleware	• Rule-based • Ontology	Static	×
CONASYS	• Rule-based	Static	×
Cooperative middleware	• Rule-based • Ontology	Static	×
AUM-IoT	• Rule-based • Ontology	Static	×

Table 3.3: Pervasive middleware comparison: Decision-making level

As revealed from Table 3.1, almost all pervasive middleware solutions [Salber et al., 1999, Román et al., 2002, Capra et al., 2003, Chen et al., 2003, Gu et al., 2004, Rouvoy et al., 2008, Wei and Chan, 2013, Celdrán et al., 2014, Evchina et al., 2015, de Matos et al., 2017, Hoque et al., 2017] that we assess do support context-awareness, are not all designed to consider adaptability issues. For example, Context-Toolkit lacks adaptability, autonomy, scalability and real-timeness to be used in an unknown and unpredictable environment. Therefore, the main expected advantage, which is intended to ensure automatic and real-time flexibility, is not available in the Context-Toolkit middleware. In contrast, some middleware could achieve a certain degree of adaptability. For example, CAMPUS shows itself in the automatic inference of adaptation decisions at runtime. However, there is still no middleware solution that could make significant progress in the last decades, particularly regarding automatic self-adaptability without human intervention at runtime. In addition, middleware solutions like CARISMA, Context-aware knowledge-based middleware and SeCoMan could achieve high scalability. Thus, they provide an efficient mechanism for application updates. Unlike, due to the limitations of their architectures, other middleware solutions like Context-Toolkit, GAIA, MobiPADS, CoBrA, SOCAM, MADAM, CONASYS, Cooperative middleware and AUM-IoT generally perform poor scalability. Apart from that, MobiPADS, CAMPUS and CONASYS can truly achieve real-timeness. They enable applications to instantly make decisions based on current or historical data. Other middleware solutions require too much additional computing, so they perform poorly in terms of real-timeness. By being aware of these limitations, we are encouraged to focus on proposing an approach to augment the capabilities of an existing middleware to reflect real-time changes occurring in dynamic environments.

In fact, as seen in Table 3.2, reviewed middleware used different context modeling for achieving context-awareness. Undoubtedly, ontology has dominated the landscape of modeling context data in the majority of investigated middleware solutions [Chen et al., 2003, Gu et al., 2004, Wei and Chan, 2013, Celdrán et al., 2014, Evchina et al., 2015, Hoque et al., 2017, Pradeep et al., 2021]. Extra criteria are enforced by the adoption of ontology such as modularity, high-level context abstraction, powerful reasoning, semantic interoperability and probably advanced context-awareness [Li et al., 2015]. However, some of these ontology-based middleware solutions [Chen et al., 2003, Celdrán et al., 2014, Evchina et al., 2015, Hoque et al., 2017, Pradeep et al., 2021] fail to provide context autonomy and evolution. Moreover, solutions, such as Music and Campus [Rouvoy et al., 2008, Wei and Chan, 2013], support higher context autonomy in a single framework. But, they did not consider the context evolution in their scope. In contrast, there are few middleware solutions that involve the context evolution. For instance, the Context-Toolkit could expand the environment by inserting new context types and entities. Similarly, SOCAM offers opportunities to introduce new entities and context types by registering new context producers. However, these entities are not dynamically introduced in the ontology. In addition to these solutions, CoBrA can accept incoming devices thanks to its context broker middleware, but it provides no way to dynamically extend this ontology to cope with incoming unknown devices. For this reason, we are motivated to work further on the expressiveness of context models' issues by using ontology models to achieve higher context-awareness and context evolution through automatically including new context features at runtime to reflect changes in users' preferences and behaviors or in surrounding dynamic environments.

Another aspect that the middleware must consider is the decision-making as demonstrated from Table 3.3. Having a deep look at the reasoning requirement, it can be found that some middleware solutions carried out limited reasoning. For instance, GAIA supports limited reasoning by using Probabilistic reasoning and Fuzzy logic. Differently, other middleware solutions, such as CoBrA, make use of ontology and rule-based inference for reasoning about high-level contextual information. As far as decision-making is concerned, different kinds of decision-making processes are reached by different middleware. Some middleware solutions, such as CARISMA, have used static decision-making where the decision process is hard-coded and its modification requires human intervention to recompile and redeploy the whole system or some of its components at runtime. Otherwise, dynamic decision-making is supported and managed by MUSIC, CAMPUS and GAIA middleware that can be changed at runtime to create or adjust the behavior of either functional or behavioral adaptations. For instance, the strength of CAMPUS shows itself in the automatic inference of context-aware adaptation decisions at runtime. But, we notice that it derives adaptation decisions according to runtime contextual information which is not quite beneficial to provide an application that could support the dynamic environment's changes. Therefore, dynamic decision-making is not well established and inaccurate in these middleware solutions. Thus, they cannot formulate a learning mechanism to generate new decision knowledge to cope with changes in their surrounding environments at runtime. Therefore, the integration of human factors into the decision-making adaptability is necessary as in the case of MUSIC middleware. These gaps, if not resolved, would lead to poor selection of decisions and services tailored to meet the real-time changes in the environment and users' preferences. Thus, how to realize the real-time and dynamic decision-making adaptation, where new decisions can be generated according to the context evolution to support intelligent context-aware solutions is also a hot topic in pervasive middleware.

Taking a holistic view on the discussed middleware solutions, it can be found that most of them only focus on a static context model. In other words, they fail in considering the automatic context model evolution at runtime. In addition, in most cases, they do not provide much importance to dynamic decision-making and only a few efforts are seen that tried to resolve this issue. However, to shift middleware solutions to dynamic environments will exert more stress on the context evolution and decision-making adaptation issues. Thus, the gaps in the surveyed middleware give us directions to focus on these specific issues, which need to be further worked on to address them. So, we put forward two important open issues including automatic context evolution and decision-making adaptation, which are among the core interests of this thesis, at runtime in middleware solutions. For that, we plan to augment an existing middleware to provide support for such automatic context evolution and decision-making adaptation at runtime to face changes in dynamic context-aware environments.

3.3 Ontology Learning Approaches

Another research line touched upon in this thesis is the context evolution that has been thoroughly investigated. When it comes to ontology-based context evolution, there are generally two methods, called ontology evolution and ontology learning, which can be addressed, since our approach for the context evolution envisions an ontology-based context model. As previously outlined in section 2.4 of the Background chapter, ontology evolution is "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artifacts" [Haase and Stojanovic, 2005], whereas ontology learning involves building an ontology from scratch, enriching, or adapting a preexisting ontology to the arisen changes [Gómez-Pérez et al., 2003] automatically or semi-automatically using structured, semi-structured or unstructured data sources [Klancnik and Blazic, 2010]. Such ontology evolution approaches do not consider data sources as an information source where ontology changes can be discovered. In this thesis, this is exactly the case, as we focus on ontology changing based on data sources. Our approach for context evolution starts with an existing ontology and supports the acquisition of new knowledge from data sources when dynamic environments or users' preferences change at runtime. Therefore, the aim of this section is the presentation of the relevant state-of-the-art for ontology learning that could offer context evolution support. This area comes closest to the context evolution contribution presented in this thesis.

3.3.1 Ontology Learning Approaches Overview

By going through the literature and several surveys in the ontology learning area [Asim et al., 2018, Lehmann and Voelker, 2014, Ma and Molnár, 2020], limited approaches have been proposed to support ontology evolution through ontology learning from data sources.

Yao et al. [Yao et al., 2014] proposed an automatic semantic extraction method to handle web data sets with semantics and generate semantic data for applications. The proposed method aims to transform semi-structured data – specifically a set of JSON documents provided by Web services – into a unified ontology. It extracts JSON data automatically, including concepts, properties, constraints and values, and merging ontologies. The resulting ontology must be validated by domain experts.

Booshehri and Luksch [Booshehri and Luksch, 2015] provided an ontology enrichment approach from text, in which Web of Linked Data, in particular, DBpedia is used. They explored DBpedia as background knowledge beside text in order to discover implicit knowledge regarding a text, from which new ontological relations, specifically object properties, are inferred. The proposed approach aimed at recommending only new object properties to ontology engineers enabling them to create much more expressive ontologies. It follows previous works in the field of ontology learning from text with the difference that it also uses the knowledge scattered in DBpedia to improve the ontology learning output.

Aggoune [Aggoune, 2018] introduced a semantic approach for automatic ontology learning from many heterogeneous RDBs in order to facilitate their integration. This approach uses WordNet as a lexical database and semantic similarity measure to help select the best terms to represent ontology components.

Kuntarto et al. [Kuntarto et al., 2019] focused on enriching Dwipa Ontology III with weather concepts from two sources of weather ontologies. These new weather concepts are generated by measuring their linguistic similarities. Authors used ontology learning that consists of several stages, mainly ontology alignment, enrichment and population.

Sbissi et al. [Sbissi et al., 2020] proposed an approach based on an automatic ontology learning from unstructured text to evolve an existing ontology. The process of ontology learning starts with the analysis of the text. Then, it switches to the extraction of relevant terminology, synonymous with the identification of terms, concepts, concept hierarchy organization, relationships, extraction of axioms. Once these elements are extracted, it updates the preexisting ontology by adding them.

3.3.2 Ontology Learning Comparative Study and Discussion

3.3.2.1 Analysis Criterion

To provide a comparative study of the discussed approaches, we introduce the following analysis criteria, which some of them are derived from previous studies [Shamsfard and Barforoush, 2003, Khadir et al., 2021]:

- **Input requirement.** This criterion is concerned with the types of inputs required by the ontology learning approaches in order to learn and evolve ontologies. Examples of such inputs are:
  - structured data, such as, existing ontologies, database schemas, or knowledge bases,
  - semi-structured data, such as, dictionaries, CSV and XML data sources,
  - unstructured data, such as texts;
- **Learned element.** This criterion is concerned with the types of learning ontological elements, such as Concept (C), Taxonomic Relation (TR), Non-Taxonomic Relation (NTR), Datatype property (D), Individual (I) and Axiom (A);
- **Pivot model.** This criterion assesses if the ontology learning approach uses a pivot model that behaves as a central piece of the evolution process to deal with inputs variability and heterogeneity;
- **Pivot model's hierarchy.** This criterion assesses if the pivot model can detect and maintain the underlying hierarchical structure in the input data;
- **Ontology Refinement.** This criterion assesses whether ontology learning approach improves learned ontologies, for example, by discovering and integrating new relations using different external resources;
- **Ontology Alignment.** This criterion is related to the process of determining correspondences between terms in a existing ontology and a learned ontology;
- **Ontology Merging.** This criterion assesses whether the discussed approaches have benefited from the merging ontology or not in order to evolve an existing ontology through the integration of new ontological elements from learned ontology;
- **Automation Degree.** The last criterion determines whether the evolution process is purely manual, semi-automatic or fully automatic. This clarifies whether human-intervention is required to assist in the evolution task at runtime.

3.3.2.2 Comparison and Discussion

In order to analyze each approach in regards with the aforementioned described criteria, we obtain a table as shown in Table 3.4.

	Input Requirement	Learned Element	Pivot Model	Pivot Model's Hierarchy	Ontology Refinement	Ontology Alignment	Ontology Merging	Automation Degree
1[1]	Semi-Structured:JSON	C+NTR+D+I	×	×	Predefined Rules	√	√	Semi-automatic
2[2]	Unstructured: Text	NTR	×	×	LOD with DBpedia	×	√	Semi-automatic
3[3]	Structured: RBD	C+NTR+D+I	×	×	×	×	√	Automatic
4[4]	Structured: Ontology	C+TR+NTR+D+I	×	×	×	√	√	Automatic
5[5]	Unstructured: Text	C+TR+NTR+D+I+A	×	×	×	×	√	Automatic

[a] [Yao et al., 2014]
[b] [Booshehri and Luksch, 2015]
[c] [Aggoune, 2018]
[d] [Kuntarto et al., 2019]
[e] [Sbissi et al., 2020]

Table 3.4: Tabular comparison of different ontology learning approaches based on the main distinguishing criteria

The presented approaches have obtained great results in evolving existing ontology using ontology learning. However, the automatic learning is a difficult task that has several important limitations with respect to the thesis objectives and research questions as pointed out in Table 3.4

First, the type of input requirements varies from one approach to another. Some approaches use unstructured texts as their input [Booshehri and Luksch, 2015, Sbissi et al., 2020], while others are concerned with structured data [Aggoune, 2018, Kuntarto et al., 2019]. However, we notice that all reviewed approaches [Yao et al., 2014, Booshehri and Luksch, 2015, Aggoune, 2018, Sbissi et al., 2020] have a single input format. For example, [Yao et al., 2014] evolved an existing ontology only from JSON data source. Unlike these approaches, we propose to handle a semi-structured input with varying formats, including CSV, JSON, and XML, and accompanied with their metadata. In addition, almost all approaches aimed to return one or a few specific learned elements. For example, [Booshehri and Luksch, 2015] extracted only non-taxonomic relations, while [Aggoune, 2018] included four learning elements, such as, concepts, non-taxonomic relations, datatype properties and individuals. In contrast to these common approaches, we cover the six different kinds of learning elements that can be appended to a learned ontology. Moreover, a one-level transformation was applied in the reviewed approaches upon a single input format, and accordingly, the pivot model' hierarchy was overlooked. Conversely, in this work, we propose a two-level transformation, since we support heterogenous formats of input. Therefore, a semi-structured input is transformed to an ontology on two levels, going through a pivot used as an intermediate model in order to represent all inputs in the same formalism. Transformation on the first level refers to the transformation of a semi-structured input to a pivot model in XML schema. On the second level, it refers to the transformation of the pivot model itself to an ontology. As far as we know, the XML pivot model's hierarchy has an important influence on the quality of the ontology alignment results since an ontology should have a hierarchy structure. For this reason, in our approach, we consider the XML hierarchy structure in pivot models by exploiting the hierarchy imposed by the XML language. Furthermore, even though the current approaches rely on the quality of the inputs to evolve ontologies, there are some approaches that consider external knowledge bases for learned ontology refinement. For example, they usually use DBpedia as by [Booshehri and Luksch, 2015] or predefined rules as by [Yao et al., 2014] to extract missing knowledge and refine learned ontologies. Regarding using these external knowledge bases separately, several limits can be listed as: the inexistence and unavailability of such dictionaries for a given domain, the limited richness of the vocabulary and supported languages. However, none of the reviewed approaches dealt with ontology refinement through coupling several methods to answer the missing knowledge during ontology learning to address these limitations. Contrary to this, we spent more effort on improving the refinement results. Therefore, LOD such as DBpedia, WordNet and lexico-syntactic patterns in metadata will be applied as references to accomplish the refinement. In contrast, we do not learn an ontology from LOD; however, we make use of LOD, WordNet and metadata as background knowledge for refining a learned ontology. Thereafter, this comparative study highlights that a range of previously discussed approaches in the scope of existing ontology evolution [Booshehri and Luksch, 2015, Aggoune, 2018, Sbissi et al., 2020] did not consider the ontology alignment, in contrast to the ontology merging. Indeed, close to [Yao et al., 2014], we consider an evolution process that is driven by the task of ontology learning. The proposed evolution process includes alignment and merging activities to consider the evolution of an existing ontology-based context model using learned elements resulting from the learned ontology. Finally, some approaches offer full automation of ontology evolution and do not involve human interaction, while others bring only a minor level of automation and rely heavily on the feedback given by the expert to improve the quality of the ontology evolution results. For purposes of our work, we explore a fully automatic way to evolve an existing ontology at runtime to cope with changes.

Following the above discussion, the approach for the context evolution described in this thesis should close gaps within the related works and focus on proposing an automatic ontology-based context model evolution approach that follows an ontology learning approach to evolve an existing ontology context model to answer the changes encountered in dynamic environments at runtime. The suggested approach for context evolution consists of five strengths:

1. fully automatic ontology-based context model evolution guided by unsupervised ontology learning once changes in the surrounding environments occur at runtime;

2. support of heterogeneous input data sources;

3. use of a hierarchical XML pivot model to maintain the input's hierarchy and to construct the learned ontology's hierarchy;

4. use of multiple pieces of background knowledge together with a pattern-based semantic analysis, including simple patterns such as regular expressions to ensure the refinement of the learned ontologies;

5. neglect of expert intervention, where the main input is an existing ontology-based context model and the main output is an evolved version of this model.

3.4 Rule Learning Approaches

Constructed rule knowledge bases at design time are far from complete and require continuous enrichment at runtime due to changes occurring in dynamic environments. Thus, there is a need for a common approach to rule knowledge base enrichment. In this sense, rule learning has attracted attention since decades, and is of particular interest nowadays. It is the most suitable approach to achieve this aim, as it plays a vital role in rule discovery from data sources. As rule learning operates in an unsupervised fashion, any elements of context which are in the data sources will automatically be included in the generated rules. By searching through the literature, there is an extensive research basis to support rule generation from data sources to tackle the challenge of automatic rule knowledge base enrichment and dynamic decision-making adaptation at runtime. In the following, we first review some approaches for generating rules within the area of context-awareness. Then, we provide a discussion to highlight the research gaps that motivate us to propose a decision-making adaptation approach.

3.4.1 Rule Learning Approaches Overview

3.4.1.1 Rule Learning Approaches Using Traditional Association Rule Mining Algorithms

A large number of rule learning approaches have been introduced using association rule mining algorithms. In this sense, we describe a few of these approaches.

Gabroveanu and Mihai [Gabroveanu and Diaconescu, 2008] proposed a recommender system for students. They used data obtained from the learning database and Apriori algorithm as an association rule mining algorithm in order to identify strong association rules for students. Then, these rules, obtained in an offline mining process, are translated into Jena rules to allow reasoning over RDF models.

Davagdorj and Ryu [Davagdorj and Ryu, 2018] offered an association rule mining method to discover useful patterns, which include medical knowledge, from a medical dataset. They applied the FP-Growth algorithm to extract a set of association rules. Then, the obtained rules are used to support medical decision-making for interpreting diagnosing patient information.

Kaliappan and Sai [Kaliappan et al., 2019] presented a new modified Apriori algorithm for finding the association rules among large datasets to promote sales and user interaction. They showed that the proposed algorithm improved the efficiency of generating association rules.

Asadianfam et al. [Asadianfam et al., 2020] introduced a new approach to improve recommendations that can be used to predict the next navigable page of users. One of the objectives considered in their approach is to provide appropriate recommendations to users who have different profiles from the existing users' profiles. To deal with the objective, authors used Apriori algorithm to generate association rules from users' behaviors and then made appropriate recommendations. They showed that the generated association rules could increase the overall efficiency of the recommender system.

Miswan et al. [Miswan et al., 2021] proposed a framework of association rule mining in readmission tasks. The proposed framework consisted of two processes, namely data pre-processing and rule mining extraction. Apriori algorithm is used to extract the hidden input variable patterns and relationships among admitted patients by generating supervised learning rules. The mined rules are discussed and validated by the domain expert, which is a valuable guide in making decisions on targeted patients' clinical resources based on various readmission durations.

3.4.1.2 Rule Learning Approaches Using Machine Learning Algorithms

Apart from these previous approaches, with the advent of Machine Learning algorithms, there are also few approaches exploring the rule learning using Machine Learning algorithms.

Hong et al. [Hong et al., 2009] proposed an agent-based framework for offering personalized services utilizing the extracting users' preference and association rules. The Decision Tree algorithm is considered to infer association rules for recommending personalized services for users.

Zulkernain et al. [Zulkernain et al., 2010] introduced an intelligent mobile interruption management system. The main idea of their proposed system is to intelligently assist users in their daily activities. To this end, a Decision Tree algorithm is used to make intelligent decisions.

Sarker et al. [Sarker, 2019] presented an association rule learning approach that can be used to discover a set of non-redundant and useful rules. In their approach, they considered, first, the Naïve Bayes (NB) algorithm to eliminate noise from data and, second, the Decision Tree algorithm to generate a set of association rules. These algorithms are used to build a robust prediction model that could improve the prediction accuracy.

Basha [Basha, 2021] provided a cardiovascular prediction system that combines the traditional K-Nearest Neighbor (KNN) algorithm with a Genetic Algorithm (GA) to extract strong association rules for facilitating the decision-making process. First, the proposed system extracts association rules using KNN algorithm. Then, the output rules become the population of the GA to remove redundant and irrelevant rules.

3.4.2 Rule Learning Approaches Discussion

Rule learning approaches have made great advances in addressing the process of generating rules for the undefined or unknown context using external data sources. They generate rules that trigger the sequential execution of actions when an undefined context feature is produced. In the following, the summary of strengths and weaknesses of the different presented rule learning approaches are illustrated in Table 3.5. As illustrated in Table 3.5, a common weakness that can be found in the majority of existing approaches mainly stands on the use of traditional association rule mining algorithms, such as FP-Growth and Apriori. This weakness is caused by the narrow applicability of these traditional algorithms due to the huge number of generated rules [Mahmood et al., 2013]. However, generating a redundant and large number of rules leads to a situation where most of the rules are in fact irrelevant, uninteresting and useless in making decisions at runtime. This redundant generation makes not only the rule-set unnecessarily large but also makes the decision-making process more complex and ineffective. Therefore, the traditional association rule mining algorithms are not able to extract interesting rules efficiently.

Additionally, the traditional approaches dealing with rule generation are not efficient anymore, which paves the way for the adoption of Machine Learning algorithms as a solution for this issue. Recently, there are many efforts, mainly around the adoption of Machine Learning algorithms in rule learning, specifically geared towards addressing the weakness arising from traditional association rule mining algorithms. As found among discussed approaches that consider Machine Learning algorithms for rule learning, Machine Learning algorithms can overcome this weakness and avoid rule redundancy. Yet even these approaches may serve as a solid solution, in many cases, approaches like [Hong et al., 2009, Zulkernain et al., 2010, Sarker, 2019, Basha, 2021] could not ensure high accuracy in generating rules [Freitas, 2000]. Furthermore, another limitation in some of these approaches [Gabroveanu and Diaconescu, 2008, Miswan et al., 2021], is that they are only suitable for offline learning and generation and not during runtime. Thus, it is witnessed that they are definitely unsuitable for dynamic environments, where new incoming changes are received continuously at runtime and may wrongly take decisions based on rules defined a priori. Apart from these limitations, a common limitation of previous approaches was that certain of them, as [Miswan et al., 2021], require domain expert intervention to validate the generated rules.

Following the above discussion, we aim to close gaps within the discussed approaches by proposing an approach for the decision-making adaptation that relies on automatically enriching an existing rule knowledge base with new decision rules generated at runtime to perform an effective decision-making in dynamic environments. The proposed approach focuses on discovering and generating non-redundant and interesting rules through the adoption of Machine Learning algorithms along with an extended GA to reduce rule redundancy. To address the accuracy question, the proposed approach represents a hybridization idea that combines the strengths of two Machine Learning algorithms. Moreover, it proceeds for a rule transformation to support the ontology-based context models and the automatic enrichment of rule knowledge bases at runtime.

	Advantages	Drawbacks
[Gabroveanu and Diaconescu, 2008]	• Translating identified rules into Jena rules to allow ontology reasoning	• Generating a redundant and large number of rules • Generating a redundant and large number of rules • Using offline learning
[Davagdorj and Ryu, 2018]	• Generating a redundant and large number of rules • Generating a redundant and large number of rules	
[Kaliappan et al., 2019]	• Presenting a modified Apriori • Improving generating association rules efficiency	• Generating a redundant and large number of rules • Generating a redundant and large number of rules
[Asadianfam et al., 2020]	• Generating a redundant and large number of rules • Generating a redundant and large number of rules	
[Miswan et al., 2021]	• Generating a redundant and large number of rules • Generating a redundant and large number of rules • Using offline learning • Requiring domain expert intervention for rule validation	
[Hong et al., 2009]	• Using Machine Learning algorithm • Avoiding rule redundancy	• Could not ensure high accuracy in generating rules
[Zulkernain et al., 2010]	• Using Machine Learning algorithm • Avoiding rule redundancy	• Could not ensure high accuracy in generating rules
[Sarker, 2019]	• Using Machine Learning algorithm • Avoiding rule redundancy • Combining NB and Decision Tree	• Could not ensure high accuracy in generating rules
[Basha, 2021]	• Using Machine Learning algorithm • Avoiding rule redundancy • Combining KNN and GA	• Could not ensure high accuracy in generating rules

Table 3.5: Advantages and drawbacks of different rule learning approaches

3.5 Concluding Remarks

This chapter comprehensively reviewed the main existing approaches in different topics that provide support for accomplishing contributions that have been defined to offer intelligent context-aware solutions in dynamic environments.

To conclude this chapter, we highlight these key findings:

- Firstly, in order to compile a comprehensive set of requirements for pervasive middleware solutions, we conducted an exhaustive middleware survey and discussion. Following this discussion, we underline that the current middleware solutions revealed a set of issues in coverage of context models and decision-making at runtime, especially in dynamic environments (see section 3.2). In chapter 4, we will present our solution to address these issues, where we will provide IConAS approach to augment such an existing middleware solution in order to support automatic context evolution as well as decision-making adaptation at runtime in dynamic environments.

- Secondly, we presented an overview of different ontology learning approaches that are used to evolve existing ontology models. Then, we introduced a comparison between these approaches based on different criteria. However, these approaches support the ontology evolution at a low level of abstraction (see section 3.3). Unlike them, in chapter 5, we attempt to confront the automatic evolution of an existing ontology-based context model at runtime, by using unsupervised ontology learning from semi-structured data sources.

- Finally, we performed a targeted literature review on rule learning approaches that are of importance for decision-making adaptation and rule knowledge base enrichment. In spite of the vast amount of the discussed literature, which has been devoted to the rule learning from semi-structured data, we acknowledge that there is still work to be done in order to completely solve the challenges of decision-making adaptation at runtime confronted in this thesis (see section 3.4). In chapter 6, we will introduce a new approach for rule learning using Machine Learning, where non-redundant and interesting rules can be automatically generated at runtime with high accuracy.

The key findings provide a basis to design and develop an intelligent context-aware solution for dynamic environments.

CHAPTER 4

IConAS Approach

4.1 Introduction

The pervasive middleware solutions in dynamic environments are experiencing major problems that connect context models and decision-making as demonstrated in chapter 1 and chapter 3. In the bid to tackle these pivotal problems, an existing middleware solution augmentation that attempts to provide a more appropriate view of the context models and decision-making in dynamic environments at runtime, is needed. In this regard, we target the first research objective:

<O>. to propose a hybrid approach that aims to augment an existing middleware solution for providing intelligent context-aware solutions. This proposed approach offers context-aware solutions the ability to automatically and dynamically evolve their context models and their decision-making when facing changes in enclosing dynamic environments or in users' preferences and behaviors at runtime, without lowest human contribution possible. This approach investigates the following research question:

RQ. *How can existing middleware solution boundaries be augmented to automatically meet changes in users' preferences and behaviors or in surrounding dynamic environments at runtime without the lowest human contribution possible?*

For fulfilling this objective and answering this research question, we undertake a hybrid approach, called Intelligent Context-Aware Solution (IConAS), with the inclusion of context evolution and decision-making adaptation approaches to face changes in dynamic environments at runtime. For that, we devote this chapter to giving an overview of IConAS approach and its architectural design. First, we introduce the IConAS approach. Second, we present IConAS's reference architecture, which provides a roadmap for the design of intelligent context-aware solutions through augmenting existing middleware solutions. Furthermore, we provide a detailed description of IConAS's reference architecture instantiation in a specific domain so as to make concrete the achievement of ***<O>.***.

This chapter is organized as follows:

- Section 4.2 presents IConAS approach overview to describe its key features and their respective functionalities.

- Section 4.3 describes the reference architecture of the IConAS approach. In addition, this section explores how the proposed reference architecture can be instantiated in order to achieve the desired context-aware solution behavior in elderly healthcare domain.

- Section 4.4 concludes this chapter by drawing together the key points of the proposed IConAS approach.

4.2 IConAS Approach Overview

We present IConAS approach that takes into consideration the entire range of problems, stated in section 1.1, to deal with dynamic environments. The essence of IConAS approach is to manage dynamically context changes in real-time toward offering intelligent context-aware solutions in dynamic environments. To achieve this, IConAS aims to augment an existing middleware to provide the possibility to add, remove or replace context features and decision rules depending on the captured context changes at runtime, thus allowing a context-aware solution not to be static, and capable of handling runtime context evolution and decision-making adaptation to reflect dynamic environments' changes without human's intervention. More precisely, every time a new context appears at runtime, the proposed IConAS approach begins with evolving the existing context model and finishes with enriching an existing rule knowledge base with new generated decision rules.

For that, IConAS approach is based on a hybrid approach in the form of a combination of two key approaches, namely: Context Evolution (CoE) approach [Jabla et al., 2021a] and the second Decision-Making Adaptation (DMA) approach [Jabla et al., 2022a, Jabla et al., 2021b, Jabla et al., 2022c], to satisfy thesis's objectives. This hybrid approach makes the architecture suitable for further expansion and adaptation. Figure 4.1 provides a simplified representation of IConAS approach to understand the both approaches they consist of.

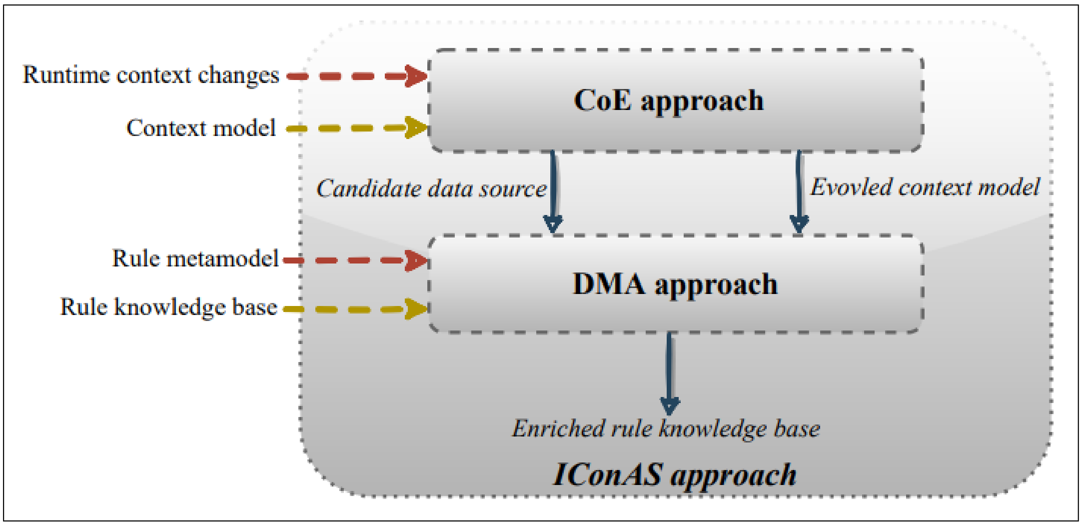

Figure 4.1: IConAS approach representation.

As shown in Figure 4.1, IConAS approach is decomposed in:

- **CoE approach** that provides support for the timely evolution of existing context models based on ontology to represent arisen context changes in dynamic environments and in the users' preferences at runtime. As shown in Figure 4.2, this approach entails different modules, such as,data source selection, data source format unification, ontology-based context learning and ontology-based context integration modules, that allow evolving an existing ontology-based context model. To meet this evolution, an example of different techniques and methods used would be ontology learning, ontology refinement, ontology alignment and ontology merging. As a result, the CoE approach offers an evolved version of an existing ontology-based context model according to environment changes detected at runtime. More details regarding the CoE approach and its key modules are provided in chapter 5;

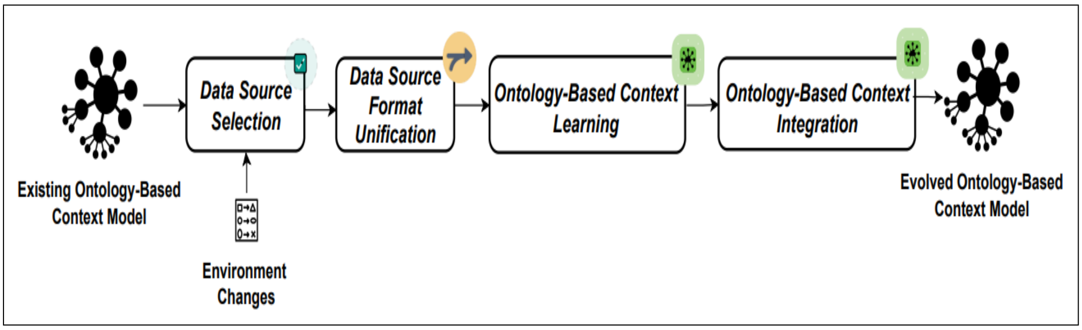

Figure 4.2: CoE approach's key modules.

- **DMA approach** that proceeds by searching for missing decision rules and allows the generation of new ones to go along with the evolved context model at runtime. To achieve this adaptation, this approach entails different modules, such as rule generation and rule transformation modules, that allow learning and generating decision rules to enrich an existing rule knowledge base as depicted in Figure 4.3. An example of algorithms used would be Machine Learning and GA. As a result, the DMA approach produces an enriched rule knowledge base. More details on the DMA approach and its modules are provided in chapter 6.

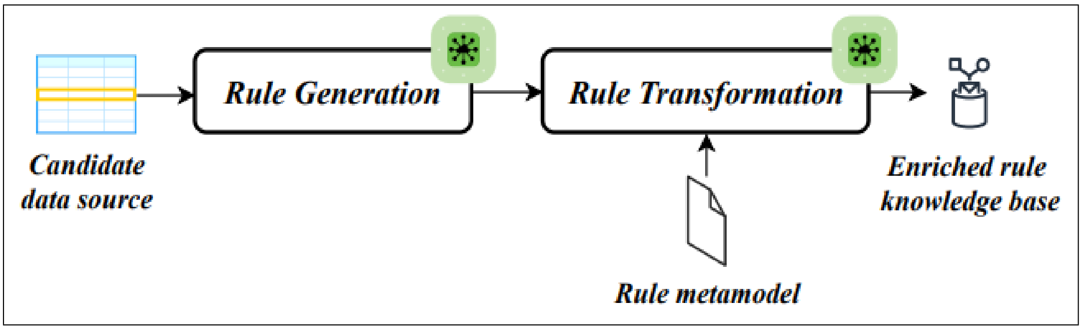

Figure 4.3: DMA approach's key modules.

4.3 IConAS Approach Design

After giving an overview of the IConAS approach, we detail IConAS' reference architecture and its instantiation in a specific domain.

4.3.1 IConAS Reference Architecture

We present IConAS' reference architecture that provides abstractions and guidelines for the specification of IConAS in a certain domain. In the following, Figure 4.4 depicts the general representation of IConAS's reference architecture.

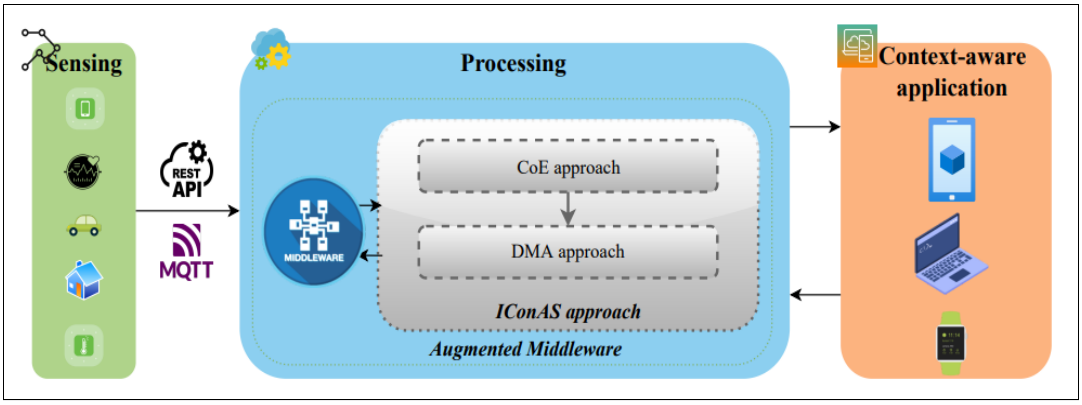

Figure 4.4: Reference architecture for the IConAS approach.

As shown in Figure 4.4, the reference architecture is a 3-layer architecture comprising Sensing, Processing and Application layers, which are described as follows.

- **Sensing layer** that serves as an interface between the physical and digital world. This process is performed by sensing sensors. It is also responsible for keeping track of users, since this layer is embedded in mobile devices, such as smartphones. Moreover, the sensor layer communicates in real-time with the processing layer. In this type of communication, the sensor layer serves as the client, which feeds the server on the other side with a continuous raw data stream;
- **Processing layer** that is the main layer of this work. It encloses an augmented existing middleware. It takes as input raw data and performs comprehensive analysis and processing of the data obtained from the sensing layer. All the analysis and processing will be briefly described in the following sections and further be deeply investigated in chapter 5 and chapter 6;
- **Application layer** that is responsible for providing information and services to users. It can be applied to many areas of healthcare, such as elderly people monitoring, through the use of a variety of sensors, such as embedded sensors in mobile devices.

4.3.2 IConAS Concrete Architecture in Healthcare Domain - IntElyCare Framework

In contrast to reference architecture that is presented to facilitate IConAS's design and development in multiple domains, concrete architecture is designed and used in a specific domain as explained by [Angelov et al., 2012]. Thus, IConAS's concrete architecture can be used to validate IConAS's reference architecture shown in Figure 4.4 in several domains ranging from smart cities to healthcare. In this sense, we provide the Intelligent Elderly Healthcare (IntElyCare) framework for deriving the IConAS's concrete architecture in

Elderly Healthcare domain [Jabla et al., 2022b]. The reasons behind this domain choice are: first of all, the growing number of elderlies living alone and in need of specialized care due to the demographic development and the aging of the population; secondly, in the era of pervasive computing, elderly living has become smarter by the latest advancements in wearable sensors and telecommunication technologies in order to deliver healthcare services. The remainder of this subsection presents the IntElyCare framework overview and details about the concrete architecture, which defines IntElyCare design.

4.3.2.1 IntElyCare Framework Overview

In response to the Elderly Healthcare domain, Human Activity Recognition (HAR) is an important application that aims to track the physical activities of elderly people using sensors for faster analysis, decision-making and better healthcare recommendations. HAR has increasingly received attention in recent years to track regular activities of elderly. However, the vast majority of existing HAR approaches considerably contributed to the understanding and analysis of elderly behavior under static and predicted environments. A key assumption in such a static approach is that they are typically built to recognize only a set of predefined activities of daily living [Lara and Labrador, 2012]. These predefined activities are not necessarily valid in dynamic environments for a longer period of time due to the diversity in human activity preferences. This diversity can lead to the occurrence of unknown activities and subsequently dramatic decrease in recognition accuracy. The reason for such accuracy decrease is that existing HAR approaches work efficiently only in controlled and static environments and could not recognize unknown activities. Therefore, it is extremely important that an approach be able to learn and recognize unknown activities upon any changes in surrounding dynamic environments at runtime, such as adoption of the application by new elderly or changes in elderly behavioral preferences. However, there are still numerous challenges confronting existing approaches since activities performed by elderly have considerable variability. Generally, this variability renders the used training or ontology models with predefined common activities unsuitable. Therefore, as elderly surrounding environment is dynamic at runtime, creating an approach that is able to dynamically leverage new and unknown activities becomes important.

For the IConAS's reference architecture instantiation, we propose a framework, named Intelligent Elderly Healthcare (IntElyCare), for elderly activity recognition and healthcare decision-making in dynamic environments using smartphones. IntElyCare framework envisions taking a knowledge-driven activity recognition approach to reinforce the recognition accuracy and elderly's quality of life from different human activity recognition data sources that are essentially collected from smartphones' sensors, such as, accelerometer and gyroscope. For that, IntElyCare considers recognizing simple human activities, such as sitting, standing, running, walking, biking and so on. In the context of dynamic environments, IntElyCare discovers known and unknown activities and then automatically learns new categories for the unknown activities to reinforce the capability of activity recognition in dynamic environments. More specifically, we propose an ontology-based context evolution along with a dynamic decision-making adaptation, so that unknown activities performed by an elderly can be automatically recognized properly at runtime. The IntElyCare framework uses existing ontology-based context model and rule knowledge base. The ontology-based context model can formally specify concepts, such as activity, sensor, location, etc., and their relations in a given context. The rule knowledge base consists of rules to predict elderly performed activities. The key idea in the present framework is to automatically evolve the ontology-based context model and enrich the rule knowledge base with new rules for learning and recognizing unknown activities at runtime. In essence, the main contributions of the proposed IntElyCare framework can be enumerated as follows:

1. Proposing a knowledge-driven activity recognition framework to handle incomplete data, such as unknown activities, in dynamic environments;
2. Introducing an automatic ontology-based context evolution based on an unsupervised ontology learning approach, to enrich the context model with new entities, such as unknown activities, from semi-structured data;
3. Presenting a dynamic decision-making to learn new decision rules from the semi-structured data source and further enrich the rule knowledge base.

4.3.2.2 IntElyCare Framework Architecture

The design of the concrete architecture is being done by referring to the reference architecture depicted in Figure 4.4 and further customized to align with the circumstances of the selected domain and the proposed IntElyCare framework. Figure 4.5 illustrates the concrete architecture, which includes more context details of the Elderly Healthcare domain.

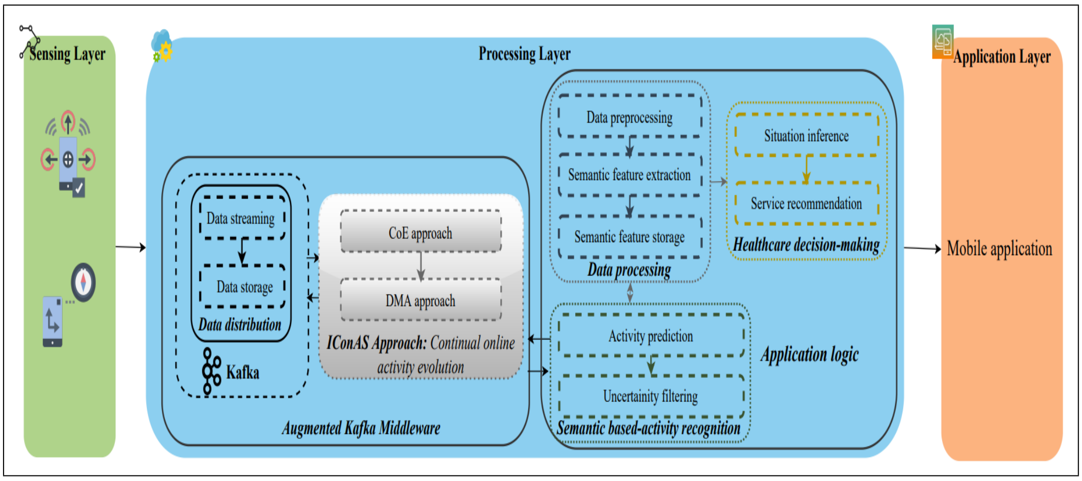

Figure 4.5: Concrete architecture for the IConAS approach - IntElyCare architecture.

As shown in Figure 4.5, the IntElyCare framework architecture consists of the three layers including several phases, such as, data processing phase, semantic-based activity recognition phase, continual online activity evolution phase and so on. The second and third phases rely on an ontology-based context model and a rule knowledge base. It is important to note that context is a very broad term as established in the Background chapter (chapter 2). In this thesis, with a context model, we mainly represent the information about the elderly performing activities and the environment, which surrounds the elderly, such as sensors, locations, time, etc. In the rest of this subsection, we briefly describe the aim and functionality of each layer.

I. **Sensing layer**

Sensing layer is at the heart of mobile crowdsensing as it describes sensors. It is the perception layer of the architecture where the inertial data is generated. It includes the sensing elements needed to acquire data from mobile devices and sampling strategies to efficiently gather it. To particularize the study with a real-life application, we consider mobile devices that usually acquire data to deal with the dynamic aspects involving contextual elements, such as activity, location and time data. A mobile device can include popular and widespread embedded sensors such as accelerometer, gyroscope, GPS, but also the latest generation sensors, such as NFC. Sensors embedded in the mobile device are fundamental for its normal utilization (e.g., accelerometer to automatically sense body movement in three directions of X, Y and Z-axis or GPS sensor to determine the person location), but can be exploited also to acquire data. More specifically, in IntElyCare framework, activity data is collected across an accelerometer sensor embedded into a user's mobile device. Data acquired through sensors is transmitted to the augmented Kafka middleware exploiting the REST API, as explained in the following Processing layer.

II. **Processing layer**

Processing layer is the key layer of the overall architecture because this is where our key contributions can take place such as evolving an existing ontology-based context model and adapting a decision-making process at runtime. Therefore, this second layer is where the bulk of the processing is performed. It is designed around two concepts, namely: Augmented Kafka middleware and Application logic. Each concept is described in further detail in turn in the following:

a) **Augmented Kafka middleware**

The augmented Kafka middleware makes use of Kafka [Garg, 2013] as a middleware solution that is well suited in our case, since it is an existing interoperable and extensible middleware solution. The augmented Kafka middleware consists of a remote server that receives collected data from the Sensing layer through REST API.

i. **Kafka middleware**

Kafka is in charge of producing, consuming and inferring the schema from the simple inertial data received. Upon arrival at the Kafka middleware, gathered inertial data are passed to the Data distribution that consists of:

A. **Data streaming**

Data streaming is responsible for creating a real-time data flow with acquired inertial data through continuously sending inertial data in real-time. Therefore, the basic functionality would be to send these data periodically to the Kafka topics using the Producer API. In Kafka, data is received in Kafka topics. A Kafka topic is a channel in which the messages are published using the Producer API. The acquired data is mostly represented as raw data as shown in Figure 4.6 for the event type "HumanActivity".

```
HumanActivity, 1642075741021, 1.5807584524154663, 9.435796737670898, 1.1668001413345337
```

Figure 4.6: Human activity event as raw data.

B. **Data storage**

Data storage is responsible for storing all the data published in the Kafka topics. It provides secure central storage to store produced data from Data streaming. For the sake of consistency and clarity, we follow a structured format to represent the acquired data in JSON. For instance, the event type “HumanActivity” in this domain will be stored in the Data storage as illustrated in Figure 4.7.

"eventTypeName": "HumanActivity",
"time": 1642075741021,
"accX": 1.5807584524154663,
"accY": 9.435796737670898,
"accZ": 1.1668001413345337

Figure 4.7: Human activity event as JSON.

ii. **IConAS approach: "Continual online activity evolution"**

Obviously, when it comes to unknown performed activities detected by the framework, and more specifically, by the Uncertainty filtering further described in II.b, we focus on activity learning for recognizing these activities, which are not covered in the existing ontology-based context model. Thus, our proposed IConAS approach performs the continual online activity evolution through the evolution of the ontology-based context model as well as the enrichment of the rule knowledge base to accurately recognize unknown activities at runtime. Figure 4.8 shows an overview of the continual online activity evolution as an illustrative example.

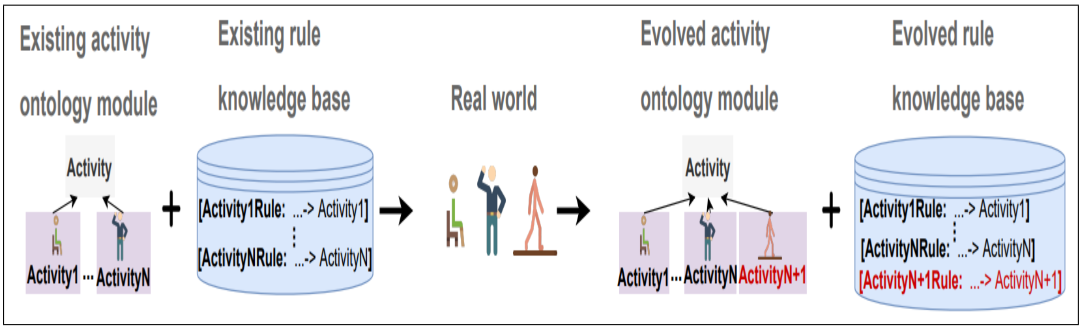

Figure 4.8: An overview of the continual online activity evolution.

In this example, the elderly starts with an ontology-based context model containing a predefined set of activities. Then, this context model evolves once unknown activities arrive at runtime. The phase of the continual online activity evolution is realized by CoE and DMA approaches previously described in section 4.3:

A. **CoE approach.**

When unknown activity instances are filtered out by the Uncertainty filtering, IntElyCare applies the CoE approach, inspired by the idea of supporting the continuous learning of the ontology-based context model at runtime. The CoE approach is applied to evolve the existing ontology-based context model by learning new activities. For fulfilling the evolution, an automatic unsupervised ontology learning from a semi-structured data source is performed. The CoE approach involves several modules, such as data source selection, data source format unification, ontology-based model learning and ontology-based model integration modules, that will be well addressed in the subsequent chapter (chapter 5).

B. **DMA approach.**
The decision-making issue is leveraged by this evolution. To address this issue, IntElyCare applies the proposed DMA approach that targets improving decision-making through automatically generating new rules to enrich the rule knowledge base under dynamic environments. The principles of the proposed DMA are (i) the creation of decision trees through decision tree machine learning algorithms and the generation of a concise set of non-redundant IF-THEN rules, (ii) the optimization of generated IF-THEN rules and (iii) the transformation of optimized IF-THEN rules to decision rules expressed in Jena for the rule knowledge base enrichment. For fulfilling these principles, the present approach involves primarily two modules: IF-THEN rule generation and decision rule transformation that will be elaborated in chapter 6.

This phase is always carried out at runtime, meaning that the IntElyCare framework does not require stopping to do so. More details of this process are available in chapter 5 and chapter 6.

b) **Application logic**
Application logic encapsulates the logic of the context-aware application and will make calls for additional resources from the augmented Kafka middleware. Therefore, it includes:

i. **Data processing**

A. **Data preprocessing**
Data preprocessing handles all the data stored in the Data storage. It focuses on performing different tasks for the cleaning of the inertial data, received in the Kafka topics, from any inconsistencies, such as, empty values, duplicates and outliers to feed the Semantic feature extraction.

B. **Semantic feature extraction**
Semantic feature extraction is first in charge of receiving the inertial data already preprocessed by the Data preprocessing. Once the preprocessed data are reached, the Semantic features extraction extracts useful features from them. Within the scope of the IntElyCare framework, these semantic features represent a few context features such as who performs the activity, when the activity takes place and what are the sensor measurements.

C. **Semantic feature storage**
Given the extracted semantic features, the existing ontology-based context model is populated and integrated in a formalism that is capable of

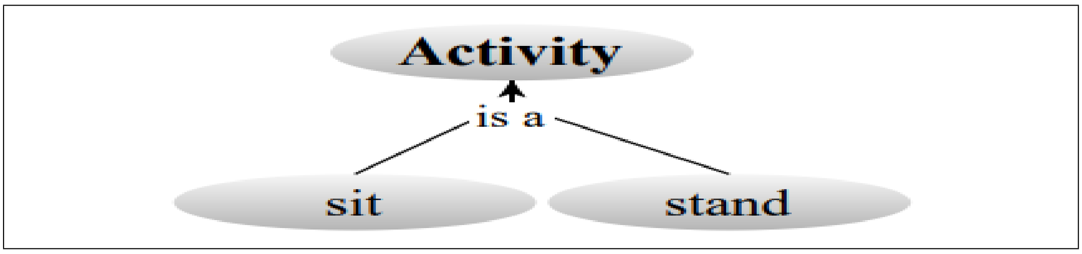

Figure 4.10: Activity ontology module with an example of a closed set of predefined activities.

reasoning the activity type of users. As known, there are several well-known ontologies that propose a formalism for context and activities, like SOUPA [Chen et al., 2004b] and MetaQ [Meditskos et al., 2016]. In the IntElyCare framework, we make use of our modular ontology-based context model, already referred to in our previous paper [Jabla et al., 2020], to abstract and generalize context features into separate ontologies for better reusability, flexibility and maintainability. This modular ontology consists of a set of interrelated ontologies, known as activity, sensor, device, situation, service, time and location ontology modules. In this thesis, we cover the following ontology modules:

- Sensor ontology module that is used to abstract and describe the attributes and performance of embedded sensors in mobile devices. It is built on top of SOSA/SSN ontology [Haller et al., 2019] to represent the relationships between sensors and their measurements. Figure 4.9 shows a partial representation of the sensor ontology module;

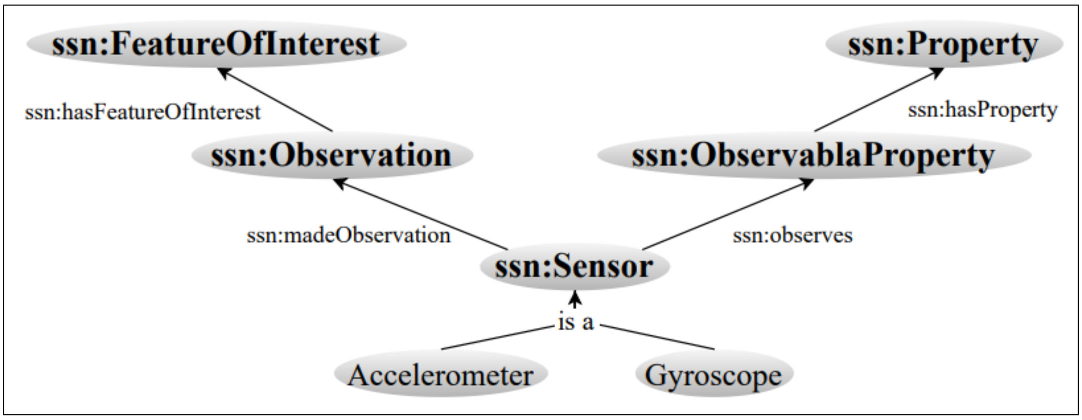

Figure 4.9: A partial view of sensor ontology module.

- Activity ontology module that is used to abstract and describe the property of daily living activities. The aim of the activity ontology is to model the different activities that can be performed by an individual and to perform the reasoning mechanism. Figure 4.10 represents the structure of activity ontology with an example of a closed set of predefined activities;
- Device ontology module that defines knowledge about devices that are used to record raw sensory data;

- Situation ontology module that contributes to identify the possible situations depending on elderly contextual information to provide relevant service selection in order to meet their needs as closely as possible. As illustrated in Figure 4.11, the situation consists of pertinent conditions that can be composed of the currently available context information to thoroughly understand elderly and improve their situation identification. A situation has different properties to characterize it, such as name, time, location and others. An elderly situation, which is a sub concept of situation, can be either a daily situation binding a normal situation or an irregular situation related to urgent situation, such as, elderly's health issues;

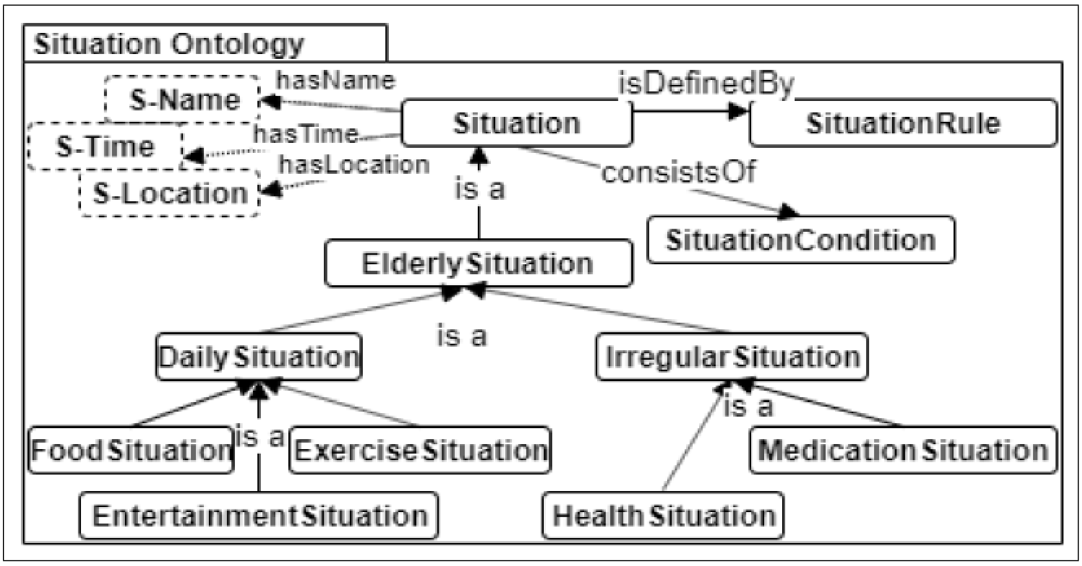

Figure 4.11: A partial view of situation ontology module.

- Service ontology module that provides a way for describing context-triggered services through the context and situation aware-based reasoning results. This sub-ontology for semantic service description adopts basic concepts and relations from a service ontology called OWL-S [Martin et al., 2004] since it is tailored to services in general along with the Web services and the semantic Web. Service ontology module expands the OWL-S ontology to include additional features, such as elderly service, elderly service profile and elderly service model, that extend, respectively, the OWL-S elements: service, service profile and service model. These elements are the core concepts of our service ontology as illustrated in Figure 4.12. An elderly service is triggered by an inferred situation. Each elderly service presents a profile to describe its characteristics by defining its name, input, output, precondition and intended purpose. Additionally, an elderly service profile can have a category. This elderly service category is divided into two basic categories: physiological elderly services and safety elderly services. Moreover, elderly service is described by a model that deals with its internal structure.

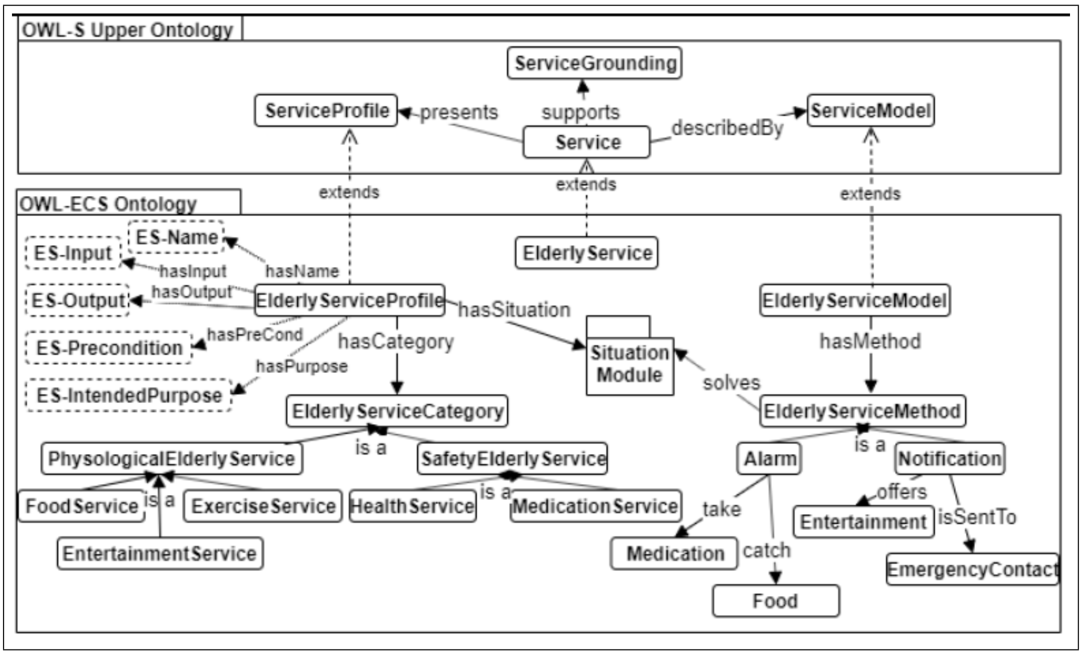

Figure 4.12: A partial view of service ontology module.

This elderly service model executes its own method, which defines the operational description related to the elderly service profile, to carry out the corresponding service;

- Elderly ontology module is a subclass of the context that represents and captures the elderly context within a changing environment. Figure 4.13 describes information about the elderly, which can alter the inferred service. An elderly has an elderly profile and an elderly constraint. Elderly profile is limited to some personal information, such as name, age, telephone, address and health status, as the elderly can be healthy or unhealthy (suffers from disease or disability). Also, it contains a medical profile that refers to the medical history of elderly including types of diseases, treatments and risk factors. As for the elderly constraint, it consists of two main branches: elderly preferences, which cover preferred entertainment content, preferred exercises (e.g., yoga, walking, biking, etc.) and preferred emergency contacts, and elderly requirements that deal with the elderly needs, such as what suggested exercises that elderly must perform;
- Time ontology module that represents the time notion in the context, which can be used to indicate the time of performed activity;
- Location ontology module that describes the location of performed activities.

ii. **Semantic-based activity recognition**

After Data processing, the Semantic-based activity recognition occurs to infer the activity types associated with the inertial data. This phase is divided into two steps:

A. **Activity prediction**

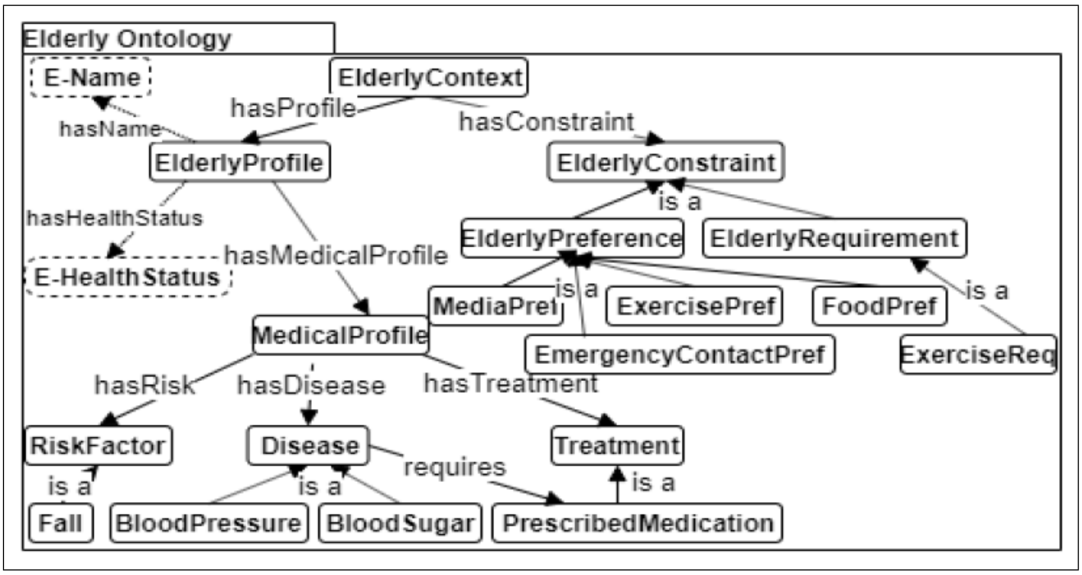

Figure 4.13: A partial view of elderly ontology module.

Once the ontology-based context model is populated with the extracted semantic features, the Activity prediction classifies the new ontology instances by processing rules in the rule knowledge base. An example of a rule for a specific activity can be depicted in Figure 4.14. Then, the activity prediction triggers queries to extract predicted activities. We assume that an ontology instance is predicted as "unknown" when it does not correlate with the predefined activities in the ontology-based context model. Stated differently, if new activities are performed, that should be recognized as "unknown".

```
Sensor(Acc) ^ CheckSup_Ax(Acc,a) ^ CheckInf_Ax(Acc,b) ^ ThresholdInf_Ax(Acc,c)
^ Ax(Acc,vx) ^ CheckInf_Ay(Acc,d) ^ ThresholdInf_Ay(Acc,e) ^ CheckSup_Ay(Acc,f)
^ Ay(Acc, vy) ^ CheckInf_Az(Acc,g) ^ ThresholdInf_Az(Acc,h) ^ CheckSup_Az(Acc,i)
^ ThresholdSup_Az(Acc,j) ^ Az(Acc,vz) ^ Activity(Act) ^ has(Act,Acc)
 ->
Name(Act, 'ActName')
```

Figure 4.14: An example of rule for an activity.

B. **Uncertainty filtering**

After performing the Activity prediction to obtain a sequence of predicted activities, the Uncertainty filtering is applied to address the recognition of performed activities, which might frequently change or even grow throughout runtime. Uncertainty filtering is in charge of processing this sequence and to identify which prediction is an unknown activity leading to "unknown" prediction. If the sequence of predicted activities contains unknown performed activities, the IntElyCare framework will proceed for the continual online activity evolution previously introduced in II.(a)ii. If not, it will continue with the next step in Healthcare decision-making. Therefore, Uncertainty filtering aims to find out uncertain predictions and to preserve the ontology instances belonging to previously unknown activities for requesting further activity recognition.

iii. **Healthcare decision-making**
The last step is dedicated to decision-making. Once the event type "Human-Activity" and the two event patterns (time, location) have been deployed in the Processing layer, it is time to feed the Application layer with data so we can recommend such services in real-time. Therefore, the goal of healthcare decision-making is to recommend to elderly users health services that are more responsive to their profiles and more aligned with their current situations at runtime. This phase is performed in two steps and represents the recommended health services as notifications.

A. **Situation inference**
Situation inference step is based on using an inference engine and rules on the available context information of an elderly for inferring their current situation in real-time to further provide them with the most relevant services. It is able to infer situations related to the elderly's context, based on data from sensors, performed activities and preferences, and taking into account the health status of each elderly;

B. **Service recommendation**
Service recommendation step sustains the provision of personalized and adapted services to go with different elderly's needs and current situations at runtime. To find and select better services for elderly, the service recommendation first determines which appropriate service should be executed through ontological inferences that are based on the current situation and elderly's profile. Second, it executes earlier selected services on the elderly's mobile device. To do so, we approach the Service recommendation by incorporating Maslow's hierarchy previously discussed in chapter 2 to offer certain assistive services for elders in the view of prior study results [Tamang, 2015]. Thus, we provide services that fulfill both Maslow's need levels that concern elderly physiological and safety needs since we need to satisfy basic healthcare needs and physiological and safety needs levels are fundamental to attain these needs. These assistive services can be arranged into two main categories to reach optimal elderly's satisfaction. The first category is elderly's physiological needs-related services and the second is elderly's safety needs-related services [Jabla et al., 2020]. In fact, both categories are often easily conflated, we investigate each of them in turn, as follows:

1) Elderly's physiological needs-related services.
In addition to the basic human physiological needs, such as food, sleep, housing, transportation, etc., elderly's physiological needs also target daily care due to age related problems. To moderate the side effects of unfulfilled physiological needs, a variety of elderly's physiological

need-related services are developed to get over their physiological barriers;

a) Exercise recommendation service.
Elderly can carry out some physical activities that require moderate efforts, such as, walking, biking, aerobics, etc., to maintain and improve their physical well-being. Hence, the increase of the effectiveness of well-being and falls-prevention needs further interventions for elderly into the behavioral patterns. To tackle that concern, we propose an exercise recommendation service that provides a reminder notification to encourage elderly to perform selected physical activity or exercise suggested by their doctors;

b) Entertainment recommendation service.
While few studies, such as [Alm et al., 2009], along these lines have revealed that the entertainment needs of the elderly people are equally important for their well-being and joyful living. For that, we provide an entertainment recommendation service that selects the relevant entertainment media and delivers notifications with the proposed media content, such as music, movies and so forth.

2) Elderly's safety needs-related services.
Once physiological requirements are met, the safety needs, such as healthcare, emergency prevention, etc., arise. Elderly's safety needs-related services look forward to covering the demand of elderly people for life safety that refers to health, emergency and medical services;

a) Health recommendation service.
Ensuring safe circumstances and protections for elderly, we aim to move when some anomalous events occur such as elderly's fall, inactivity or activeness. To accomplish this aim, a health recommendation service could raise alerts to an emergency contact, such as doctor, caregiver or family member when elderly users are falling to the ground or have not risen from bed for a long period in order to respond to an emergency event in a fast way. In the activeness of elderly, the health recommendation service could raise notifications to remind elderly users that they must go and rest for a while;

b) Medication recommendation service.
With age-related decline of memory and cognitive functionalities [Small et al., 1999], elderly may forget to take the relevant medications at the appropriate times. For this attend, a medication recommendation service is offered to provide basic medical attention for people in old age. It yields an alert to remind them about their medicines at a pre-scheduled time to experience a healthy aging.

III. **Application layer**

The top-most layer of proposed IntElyCare architecture is the Application layer. The latter contains mobile applications that can provide a service to the elderly regarding the recognized activity and situation. Thus, application provides a user interface to present the output data and services requested by the elderly. The Application layer has several functions to ensure communications between the Processing layer and the elderly. The functions in this layer are responsible for representing the incoming data to elderly in a more pleasant and human-readable format. This data is referring to predicted activities and selected services that are coming from the Processing layer. So, a pure text notification is preferred to notify the elderly for viewing the results without any user-interface interaction.

4.4 Concluding Remarks

IConAS approach is based on a hybrid approach that offers intelligent context-aware solutions through augmenting existing middleware solutions to deal with dynamic environments at runtime. The strength of the IConAS approach relies on its main approaches, which are: (1) CoE approach for evolving existing ontology-based context models to cope with runtime changes occurring in surrounding dynamic environments; (2) DMA approach for generating decision rules that reflect runtime changes and evolved context models.

Therefore, in this chapter, we have explained three stages of the IConAS approach development process:

- First, we have given an overview of IConAS approach;
- First, we have presented the reference architecture that increases understanding as an overall picture by containing typical functionality and data flows in the proposed IConAS approach;
- Second, we have provided the concrete architecture, which represents the instantiation of the presented reference architecture in elderly healthcare. Additionally, the demonstration of how this concrete architecture is instantiated – have been described in detail.

In the next chapters, we provide a detailed description of the CoE and DMA approaches that are included in the IConAS approach. First, chapter 5 investigates the CoE approach and conducts a case study. Then, chapter 6 describes the DMA approach as well as a case study.

CHAPTER 5

Context Evolution Approach

5.1 Introduction

One of the most expensive activities in a context-aware solution is the evolution of context models. As revealed by chapter 1 and chapter 3, this problem might be more severe in dynamic environments, where context may change at runtime. One of possible ways of reducing this cost is automation of the context model evolution process. Unfortunately, automated context model evolution at runtime is not as straightforward. There is a considerable challenge for any automated context model evolution approach to generate an effective evolved context model, not merely a 'structurally valid' context model. An effectively evolved context model must not only be correctly formed, according to structural modeling, it must also include semantics that ties it to new context changes that appear at runtime. In this chapter, we introduce the (CoE) approach, which forms the first part of the IConAS approach, to address this problem of effectiveness in automated evolution of context models. In this approach, the need for manual setting is no longer required by leveraging the data and information that can be acquired from data sources and external knowledge bases. For that, we target the first sub-objective:

<O.1>. To propose an approach to automatically evolve and extend context models at runtime to answer users' mobility and frequent changes in the surroundings, where we investigate the following research questions:

RQ.1. *How can the context evolution affected by users' preference and behaviors or enclosing environments' changes be automatically and dynamically supported at runtime?*

- ***RQ.1.1.*** *What kind of context modeling approach should be followed to meet an evolution of context models at runtime?*
- ***RQ.1.2.*** *How to take advantage of external knowledge bases to evolve context models?*

The rest of this chapter is organized as follows. Section 5.2 gives a general overview of the proposed CoE approach. Section 5.3 presents the CoE architecture in discussing the context evolution modules and the details of each of them for evolving an ontology-based context model to cope with context changes at runtime. Section 5.4 presents implementation details and a case study to demonstrate how the implementation can be utilized to automatically evolve ontology-based context models. Section 5.5 ends this chapter with a few concluding remarks.

5.2 CoE Approach Overview

As far as mentioned in previous chapters, expecting, at design time, how a context model evolves is challenging as we do not know a priori the changes in the surrounding environments or in the users' preferences and behaviors that will arise at runtime, especially in dynamic environments. At runtime, changes may result in invalid context for context-aware solutions. Additionally, runtime changes in the surrounding dynamic environment may affect the solution's ability to recognize a certain context due to problems or changes. Therefore, we posit that context-aware solutions need to support the evolution of their context models at runtime since the ability to embed support for the context evolution at design time is restricted.

In the light of this need, we call for runtime support for the context evolution in context-aware solutions. As much as we welcome the use of ontology, we present the CoE approach based on ontology context models [Jabla et al., 2021a]. The main challenge of this approach is the automatic evolution of an existing ontology-based context model at runtime to deal with the continuous changes of or within the surrounding dynamic environments. To meet this challenge, we exploit unsupervised ontology learning approaches for automating ontology context model evolution to some extent. In the context of this thesis, it is worth considering the idea of ontology modularity, where semantic enrichment and evolution can come into play. Here and within the modularity notion, the context model grounded on ontology could semantically be enriched and populated in a flexible way across the runtime environment changes. Through unsupervised ontology learning, the present approach underpins the evolution of an existing modular ontology context model, which acts as the initial context model, from a semi-structured data source that could match the arisen changes to conclude in answering to the runtime environment changes. We pursue evolving an initial ontology context model through transforming a semi-structured data source to an ontology, since data sources could well represent the changes arising in the enclosing environments rather than external knowledge sources, such as, DBpedia, WordNet and so on. To exhibit a strong evolution, our proposed approach is not limited to these external knowledge sources and relies on the use of semi-structured data sources, which can be far more extensive and representative for answering dynamic environments and decision-making adaptation as shown in chapter 4 and chapter 6, respectively. Generally speaking, our approach intends to be generic enough to be further applicable to any instance of semi-structured data input, i.e., the popular eXtensible Markup Language (XML), JSON and so forth. For that, we fulfill two-level transformations from semi-structured data to XML and from XML to Web Ontology Language (OWL). The first one handles data hierarchy, data type specifications [Liu et al., 2014] and the second moves backwards to the target format. The first part starts with semi-structured data, especially tabular data like CSV to XML transformation and the second part starts with XML to OWL transformation.

5.3 CoE Approach Architecture

For fulfilling an automatic context evolution at runtime, the presented CoE approach involves several modules, such as data source selection, data source format unification, ontology context model learning, and ontology context model integration modules as illustrated in Figure 5.1. In the following, we describe each module of the CoE approach from data source selection to ontology context model integration in more detail.

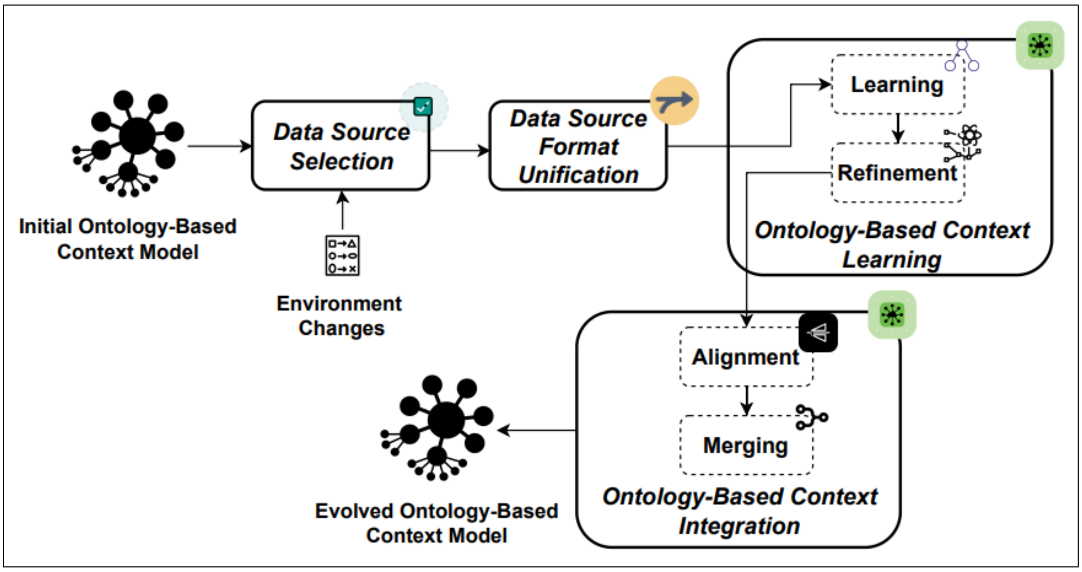

Figure 5.1: CoE approach architecture.

5.3.1 Data Source Selection

Once context changes, such as moving of a person from one location to another, sudden unavailability of devices, necessary activities and more, occur in the surrounding dynamic environment at runtime, the data source selection module aims to retrieve and select the relevant candidate semi-structured data source that can cope with the runtime captured context changes. The present module is handled by an automated data source search engine over a data sources repository. The latter is a public repository, which is populated with different semi-structured data sources and their metadata. Figure 5.2 illustrates a simple and general view of the proposed search engine.

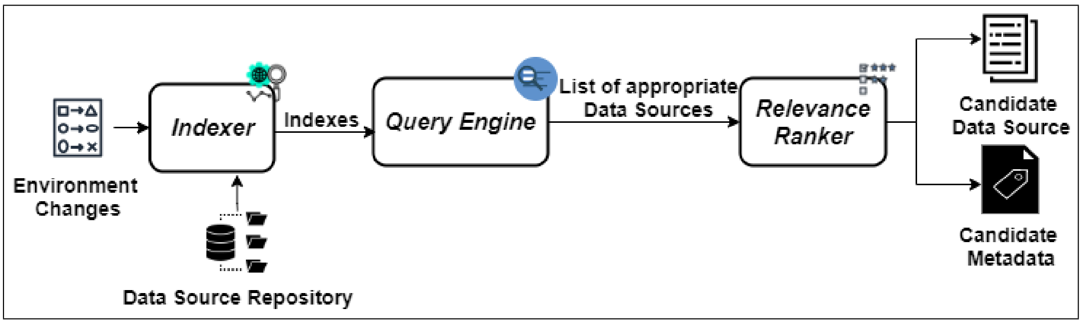

Figure 5.2: Data source search engine.

As shown in this figure, the proposed search engine takes the semi-structured data sources repository and the captured context change keywords as an input. Then, a text search is applied to the input repository to retrieve the list of the appropriate data sources that are associated with the context change keywords. Finally, we keep the relevant data source with the highest term frequency from the retrieved list to start the process of evolving the targeted ontology context model. If there are no retrieved results, the user is asked to enter an initial keyword to start the process. In the following, we put forth the theoretical basis of the main steps of the proposed search engine:

- ***An indexer*** is in charge of parsing data sources and runtime context changes to be more efficient and accurate for performing retrieval techniques. It usually performs lowercase transformation, stop words removal and stemming. Then, it stores parsed data sources in indexes;

- ***A query engine*** is responsible for formulating a query from the parsed context changes and then running it on the previously built indexes to get a list of the appropriate data sources that can match with the context changes according to their term frequencies. It allows for data sources scoring by counting the term frequency for a desired set of keywords in each data source. A very common and traditional scoring method, called Term Frequency/Inverted Document Frequency (TF-IDF) is applied to improve the relevancy of retrieval results. The term frequency "TF" is a number of times a keyword occurs in a data source. Weights are assigned to keywords in a data source based on the number of times they occur in the data source. As the TF efficiency is affected by common words like "is", "the", "a", though this limitation can be overcome by the IDF, which calculates the number of data sources that contain each keyword and reduces the weight of terms that occur in many data sources;

- ***A relevance ranker*** is in charge of ranking the retrieved list to select the most useful data source regarding the context changes. It first arranges the appropriate data sources in descending order with most relevant listed first by comparing their term frequencies and then selects the relevant data source with the highest frequency. The highly relevant data source output should be accompanied by its metadata to go further in the refinement of the learned ontology. The use of metadata is needed to increase the quality of the learned ontology by finding frequently occurring patterns and recognizable structures within the textual description of the candidate data source, since a semi-structured data source may suffer from a slight lack of semantics.

5.3.2 Data Source Unification

The main purpose of the data source unification module is to transform a candidate semi-structured data source to hierarchical XML data as a pivot model in accordance with the first-level transformation. Therefore, after selecting the candidate data source, the candidate data source is received, as CSV format file and converted to a pivot model, which is an XML data, including an XML document along with its XML Schema Definition (XSD) to handle complex hierarchy structures and highly interconnected data. Initially, the preprocessing step is started. This step takes the data source in question as the input and stops when it is cleaned and arranged in a structured form. Then, the arranged data are being parsed in order to automatically identify their schema. After that, a deep visit is performed on the parsed data to map them to the corresponding XML data by choosing an appropriate mapping rule for each tabular element and generating the corresponding XSD element. In order to achieve this, we defined a set of mapping rules, detailed in Table 5.1, for transforming Tabular Data (DT), specifically CSV data, to hierarchical XML data. We change to a new set of mapping rules since existing libraries and tools map a TD to flat XML data, which can give rise to difficulties while transforming XML data to an ontology. At the end, this module produces as the output an XML document along with an XSD document to be served as the input for the ontology-based context model's learning module.

CSV	XSD
Table name (group of CSV files)	Complex type
Column (non-nominal datatype)	Attribute with column datatype as type
Column (nominal datatype)	Element inside an anonymous complex type
Column (group of CSV files)	Element inside an anonymous complex type
Label column	Complex type uses extension

Table 5.1: TD to XSD mapping rules.

5.3.3 Ontology-based Context Learning

After performing the first level-transformation, the ontology-based model learning module takes place to apply the second-level transformation and different refinement methods. Thus, this module starts with the definition of a local ontology in the OWL language, which is discovered from the XML pivot model, and finishes with some refinements in the local ontology. Hence, both phases are involved: the learning phase and the refinement phase.

5.3.3.1 Learning Phase

The learning phase deals with the acquisition of information needed to learn an OWL ontology, which acts as a local ontology, starting from the XML pivot model. It intends to automatically transform XML to a local ontology through capturing the implicit semantics existing in the structure of XML documents. The details of this phase are described in Algorithm 5.1, where the beginning step is intended to parse the earlier generated XML schema for inspecting its features, such as complex types, elements, attributes and so on. Then, the parsed schema results are traversed to transform the pivot model to an OWL local ontology. The implementation of the following transformation adheres to certain mapping rules, which are introduced in Table 5.2. These mapping rules guide the automatic generation of a local ontology from an XML. They determine how to convert each feature of the XML schema to a semantically corresponding ontology element. For instance, xsd:complexType features are transformed to owl:Class elements, while xsd:attribute features are transformed to owl:DataProperty elements as shown in Table 5.2. Once this transformation is established using our mapping rules, the XML schema is compiled into a set of concepts, object properties and datatype properties to generate the local ontology. After that, data carried by the XML document is transformed to ontology individuals to populate the local ontology as lines from 6 to 9 in the algorithm 5.1 shows.

input: XML data, including XSD and XML documents, XMLR
  Parsed_XSD = XSDParsing(XSD document)
  do
    Generate OWLElement based on the XMLR for the XSDFeature
    LocalOntology = LocalOntology + Generated OWLElement
  until XSDFeature $\notin Parsed$
  do
    Generate OWLIndividuals
    Local_Ontology = Local_Ontology + Generated OWLIndividuals
  until XMLRow $\notin XMLdocument$
  return Local ontology

Algorithm 5.1: XML2OWL Algorithm

XML schema	OWL schema
xsd:complexType: xsd:element, containing other elements or having at least one attribute	owl:Class, coupled with owl:ObjectProperty
xsd:attribute	owl:DataProperty with a range depending on the attribute type.
xsd:element, inside an anonymous complex type.	owl:ObjectProperty
xsd:complexType, that uses extensions.	owl:Class as an owl:subClassOf “base type”

Table 5.2: XML Schema to Ontology Schema Mapping Rules.

5.3.3.2 Refinement Phase

After attaining the local ontology, it is necessary to expand its quality borders through the accreditation of refinement. Therefore, we propose three ontology refinement methods to uncover missing concepts, relations and identify concept hierarchy. For local ontology refinement purposes, DBpedia, WordNet, and metadata knowledge bases are considered as references since they are widely used and cover broad resources from different domains. Indeed, coupling several knowledge bases could answer missing knowledge during ontology refinement and improve the inexpressiveness of the local ontology through exploiting the different linguistic patterns, hyponym-hypernym relationships contained in WordNet, and concept hierarchy retrieved from DBpedia. In the following, we present these proposed refinement methods in different subsections due to their complexity.

1. ***Missing concepts refinement method***

   The first refinement method, called “missing concepts refinement”, is used to compensate for the missing concepts in the local ontology from the learning phase. It identifies the corresponding classes from the candidate data source metadata, which encompasses unstructured text providing readme file information. This identification is performed through Lexico-Syntactic Patterns (LSPs), presented in Table 5.3, which are issued from different works published in the literature [Aguado de Cea et al., 2008, Almuhareb, 2006, Sowa, 1999] to perform a semantic analysis on any textual metadata and to identify missing concepts through the datatype properties. More specifically to achieve this, this method follows the main steps outlined in Algorithm 5.2. In this algorithm, the starting step is intended to collect all datatype properties included in the local ontology. Next, the retrieved datatype properties are traversed to extract their corresponding domains. Then, the candidate metadata are processed for each datatype property sequentially in three steps. First, each sentence that contains the datatype property is extracted. Second, Parts of Speech (POS) tags are induced over each extracted sentence to prepare them for analysis. Third, a collection of LSPs is exploited over the POS tags to extract the related domain to the datatype property. Once these steps are fulfilled, missing datatype property's domains are updated to handle the missing concepts.

LSP	Example	Relation
Property|ies | characteristic|s | attribute|s of NP<class> be [PARA] [(NP<property>,) * and] NP<property>	Attributes of an accelerometer are X, Y, and Z	Datatype properties: X, Y, Z Class or domain: accelerometer
NP<class> be [(AP<property>,) *] and AP<property>	Metals are lustrous, malleable, and good conductors of heat and electricity	Datatype property: lustrous Class or domain: metal
NP<class> have NP<class>	A car has a color	Datatype property: color Class or domain: car
NN with|without DT? RB? JJ? ATTR	A pizza with some cheese.	Datatype property: cheese Class or domain: pizza
DT ATTR of DT? RB? JJ? NN	The color of the car	Datatype property: color Class or domain: car
DT NN's RB? JJ? ATTR	The car's color	Datatype property: color

Table 5.3: LSPs for missing concept acquisition.

2. ***Taxonomic relation refinement method***

The second refinement method, called "taxonomic relation refinement", identifies the concept hierarchy to organize learned and new covered concepts into the local ontology through the introduction of new subsumption relations and taxonomic structure between concept pairs. Therefore, this method focuses on exploring the benefits from coupling WordNet, metadata, and DBpedia to learn new subsumption or taxonomic relations among the concepts represented in the partially refined local ontology. Indeed, the most novel idea in this method is to compose several steps to improve the performance of the concept hierarchy refinement by taking into consideration (1) the verb hyponym and hypernym relationships contained in WordNet, (2) the behavior of different linguistic patterns by extracting hyponym-hypernym pairs from the metadata accompanied by the candidate data source, and (3) the concept hierarchy retrieved from DBpedia predicates reported in Table 5.4. Another possibility

```
input: LocalOntology, Metadata, LSPs
   DataTypePropertiesSet = GetAllDatatypeProperties (LocalOntology)
   do
      Domain = GetDomain (DatatypeProperty)
      SentencesSet = FindSentences (DatatypeProperty, Metadata)
      SentencesSet = FindSentences (DatatypeProperty, Metadata)
      do
         Sentence = PartsOfSpeechTagging(Sentence, POS)
         NewDomain = matches (Sentence, LSPs)

         if thenNewDomain ≠ ∅ then
PartiallyRefined_LocalOntology = UpdateMissingConcept (DatatypeProperty,Domain,NewDomain)
         fi
      until Sentence ∉ SentencesSet
   until DatatypeProperty ∉ DataTypePropertiesSet
   return PartiallyRefined_LocalOntology
```

Algorithm 5.2: The Proposed Missing Concepts Refinement Algorithm

to maximize the performance of the taxonomic relation refinement from the metadata is to combine the well-known Hearst [Hearst, 1992, Hearst, 1998] and Aguado de Cea [Aguado de Cea et al., 2008] patterns for hyponym and hypernym extraction, given that this combination has been proven to improve the precision and recall, since Hearst patterns allow for finding all possible taxonomic relationships with high precision but low recall, and in contrast, Aguado de Cea patterns produce high recall but low precision [Cederberg and Widdows, 2003]. These patterns, inferred from the study of linguistic patterns, can capture different semantic relations, chiefly the hyponym/hypernym relationship. Additionally, they occur frequently across text and summarize the most common ways of expressing hyponyms and hypernyms. Table 5.4 and 5.6 show a part of Hearst and Cea's patterns that are used to acquire the missing taxonomic relations between concepts in the partially refined local ontology.The proposed "taxonomic relation refinement" method is outlined in Figure 5.3, where the basic steps are displayed. At the initial step of this method, the candidate metadata is scanned for instances of distinguished Hearst and Cea patterns that are useful for detecting hyponym and hypernym relations for each concept included in the partially refined local ontology to identify new assumption relations. These patterns occurred frequently across the textual metadata and summarized the most common ways of expressing hyponyms and hypernyms. An example of these patterns that could be detected in a sentence like "Activities such as changing clothes, having guests, or cleaning are considered" is "NP , such as NP, * and|or NP", where the first NP denotes a super concept (e.g., "activity") of the next NPs (e.g., "changing clothes", "having guests", and "cleaning"). Then, a set of hyponyms and hypernyms covered by WordNet is gathered for each concept. Finally, this method finishes by checking each concept for the DBpedia predicates, such as "rdfs:subClassOf", "umbel:superClassOf", or "geo-ont:parentFeature", to discover unrecognized assumption relations among concepts using DBpedia.

DBpedia Predicate
rdfs:subClassOf
umbel:superClassOf
geo-ont:parentFeature

Table 5.4: DBpedia Predicates for taxonomic relations.

Pattern	Example	Relation
Such NP as {NP,} * {and|or} NP	Such fruit as oranges	Hyponym fruit, orange
NP {,} such as {NP,} * {and|or} NP	Vehicle such as car	Hyponym {vehicle, car}
NP {,} including {NP,} * {and|or} NP	Pets including turtles	Hyponym {turtle, pet}
NP {,} especially {NP,} * {and|or} NP	Publications especially papers	Hyponym {publication, paper}
NP {,} {and|or} other NP	Running and other activities	Hyponym {running, activity}

Table 5.5: Hearst's patterns for hyponym relation extraction with examples.

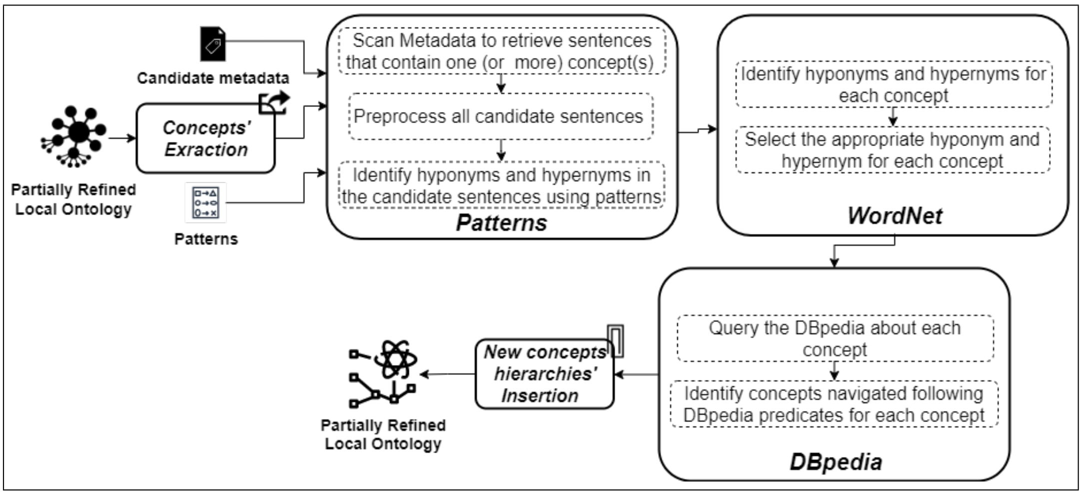

Figure 5.3: Taxonomic relation refinement method.

3. ***Non-taxonomic relation refinement method***

The third and last refinement method, called "non-taxonomic relation refinement", is based on associating concepts by representing hidden connections between them and identifying their non-taxonomic relations, specifically the object properties. This method focuses on investigating the DBpedia and the metadata as background knowledge for automatically discovering and labeling the non-taxonomic relationships to refine the inexpressive concepts existing in the partially refined local ontology, as depicted in Figure 5.4. Within this method, the DBpedia and the metadata are considered to find new non-taxonomic relations related to different concepts through analyzing the structure and dependencies of candidate sentences. First, in DBpedia, comments about every partially refined local ontology's concept are found under

Pattern	Example	Relation
[(NP<subclass>,) * and] NP<subclass> be [CN] NP<superclass>	GPS is a sensor	Hyponym {sensor, GPS}
[(NP<subclass>,) * and] NP<subclass> (classify as) | (group in|into|as) | (fall into) | (belong to) [CN] NP<superclass>	Thyroid medicines belong to the general group of hormones medicines.	Hyponym {hormone, thyroid}
There are CD | QUAN [CN] NP<superclass> PARA [(NP<subclass>,) * and] NP<subclass>	There are several kinds of memory: short term memory, and long-term memory	Hyponym {memory, short term memory} Hyponym memory, long-term memory
[A(n) | QUAN] example | examples | [CN] of NP<superclass> be | include | comprise [(NP<subclass>,) * and] NP<subclass>	Some examples of peripherals are keyboards, mice, monitors, …	Hyponym {peripheral, keyboard}
NP<superclass> be | CATV [either] NP<subclass> or | and NP<subclass>	Animals are either vertebrates or invertebrates.	Hyponym {animal, vertebrate}
NP<superclass> be divide | split | separate | group in|into [either] [(NP<subclass >,) * and] NP<subclass>	Sensors are divided into two groups: contact and non-contact sensors.	Hyponym {sensor, contact sensor}

Table 5.6: Aguado de Cea's Text Patterns for hyponym relation extraction with examples.

"rdfs:comment". Accordingly, the metadata is looped to extract all the sentences containing any partially refined local ontology concept. Then, a semantic analysis of the candidate comments and sentences is performed to locate their main components, such as, nouns, verbs, and so on. For this purpose, the candidate comments and sentences are analyzed with the aid of POS tags to identify verbs that will be used to label non-taxonomic relations.

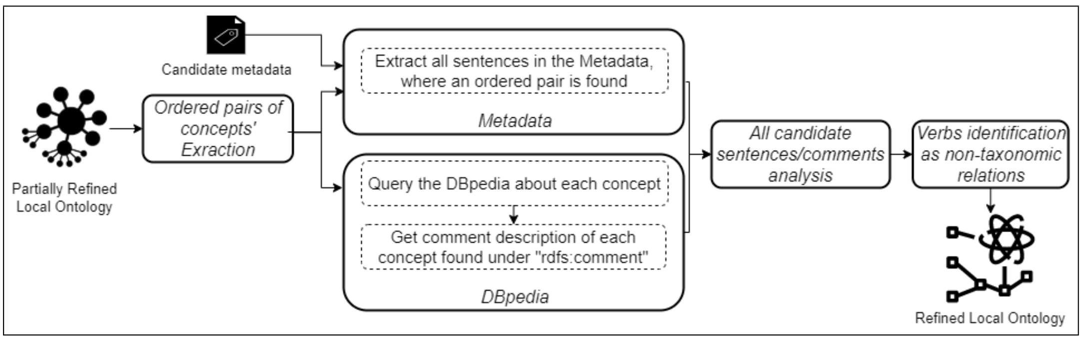

Figure 5.4: Non-taxonomic relation refinement method.

5.3.4 Ontology-based Context Integration

The ontology-based model integration module is targeted toward updating an initial ontology-based context model through a refined local ontology to answer the arisen changes in the surrounding environment. To fulfill the integration, the present module takes as inputs the two ontology models and carries out two important phases. The purpose of the first phase named alignment is to discover correspondences between the different ontologies based on the similarity calculation. The purpose of the second and last phase named merging is to merge the new entities into the initial ontology-based context model in order to obtain an evolved ontology-based context model based on the identified correspondences. In the following subsections, we introduce the two-phase process in detail.

5.3.4.1 Alignment Phase

To handle the ambiguity associated with formalization before the evolution of initial ontology-based context model evolution, it is essential to discover the correspondences between the refined local ontology and the initial ontology-based context model. To account for this, we turn to ontology alignment strategies that stand for creating links between two different ontology models through identifying semantically matching entities, which can be concepts, relations, properties, or individuals. For this purpose, we propose an automatic ontology alignment strategy considering both the syntactic and semantic similarity measures. Both similarity measures are integrated to approach the logical consistency notion. Logical consistency guarantees that the initial ontology-based context model evolution does not produce any contradictory knowledge. Therefore, the proposed strategy takes both ontology models as input and determines as output the alignment result between entities of the input ontology models by computing the syntactic similarity [Stoilos et al., 2005] followed by the semantic (or linguistic) similarity [Miller, 1995]. In the first similarity measure, a simple distance computation between two strings labeling two entities is performed. Secondly, the semantic similarity is used to compute the extent of similarity between the two entities regarding the likeliness of their meaning. This opens the door to exploring WordNet as a way of portraying semantic similarities, since WordNet can determine the semantic distance between two entities' names by considering linguistic relations such as synonym, hypernym and hyponym. Given two entities E1 and E2, their similarity sim (E1, E2) is calculated according to Equation 5.1:

$$sim(E1, E2) = \begin{cases} 1, & \text{if the entity E1 is part of the synset of the entity E2 or vice versa} \\ 0.5, & \text{if E1 is a hypernym or a hyponym of E2 or vice versa} \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

In this work, the combination of both syntactic similarity and the semantic similarity measures intends to improve the overall alignment performance. At the end, the alignment phase provides a set of matches between the different entities extracted from the refined local ontology and the initial ontology-based context model according to the chosen strategy. This matching is formally defined as (see Equation 5.2):

$$Align(O1, O2) = \begin{cases} (E1, E2, r) \\ \text{E1 } \epsilon \text{ O1,E2 } \epsilon \text{ O2} \\ \text{r } \epsilon \text{ \{equivalence,generalization,specialization,disjointness\}} \end{cases} \tag{5.2}$$

Each triplet (E1,E2,r) in Align(O1,O2) represents that entity E1 in O1 is aligned to entity E2 in O2 with the alignment relation r. The alignment relation can be:

- ***Equivalence ($\equiv$).***either entities are equal or one of them is a synonym of the other. (E1 is equivalent to E2, e.g., Person $\equiv$ Person).
- ***Specialization ($\subset$).*** if an entity or one of its synonyms is a hyponym to another entity or its synonyms. (E1 is a sub entity of E2, e.g., Author $\subset$ Person).
- ***Generalization ($\supset$).*** if an entity or one of its synonyms is a hypernym to another entity or its synonyms. (E1 is a super entity of E2, e.g., Group $\supset$ Person).
- ***Disjointness ($\perp$).*** if both entities or their synonyms are antonyms or different hyponyms of the same synset. (E1 is a disjoint entity from E2, e.g., Person $\perp$ Place).

5.3.4.2 Merging Phase

After accurately aligning the entities of the initial and refined ontology-based context models, a merging phase is performed to merge the refined local ontology with the initial ontology-based context model as depicted in Figure 5.6. In our work, the merging phase is seen as the process that updates the initial ontology-based context model through the addition of new entities such as concepts, properties, relations, and individuals from the refined local ontology to obtain a more complete ontology that can cover the environment's changes emerging at runtime. In other words, the initial ontology-based context model is enriched and populated using the refined local ontology, taking into account the set of matchings defined in the previous phase. Based on the found alignments, similar entities are merged into a single one in the initial ontology-based context model, whereas the entities considered dissimilar are directly copied into the initial ontology-based context model. More specifically, we utilize a list of merge requirements that should be met in order to evolve the initial ontology-based context model. Overall, we classify the merging requirements according to three aspects.

The three aspects we identified are:

- ***Completeness.*** refers to knowledge preservation and coverage:
  - ***R1.Concept preservation.***Each concept in the refined local ontology should be present in the initial ontology-based context model.
  - ***R2. Property preservation.*** Each property from the refined local ontology is acquired by the initial ontology-based context model.
  - ***R3. Instance preservation.*** All instances of the refined local ontology should be preserved in the initial ontology-based context model.

  - ***R4. Correspondence preservation*** If two entities of the refined local ontology and the initial ontology-based context model are equivalent, both should be merged into a single one in the initial ontology-based context model.

- ***Deduction.*** refers to the deduction satisfaction with R5:

  - ***R5. Entailment deduction satisfaction.*** The initial ontology-based context model is desirable to be able to entail all entailments of the refined local ontology. This ensures that every aspect of the refined local ontology is present in the initial ontology-based context model.

- ***Constraint.*** reflects the satisfaction of the ontology constraints:

  - ***R6. One type restriction.*** Two corresponding entities should follow the same data type [Pottinger and Bernstein, 2003]; e.g., if the range of author Id in one of the input ontologies is String and in the other one is Integer, then the range of the merged entity author Id in the merged ontology cannot have both types.
  - ***R7. Property value's constraint.*** If the (all/target) input ontologies place some restriction on a property's values (e.g., in terms of cardinality or by enumerating possible values) this should be preserved without conflict in the merged ontology [Pottinger and Bernstein, 2003].
  - ***R8. Property's domain and range oneness.*** The merge process should not result in multiple domains or ranges defined for a single property. This rule is recast from the ontology modeling issues in [Poveda-Villalón et al., 2012].

Finally, an evolved view of the initial ontology-based context model is computed from the refined local ontology.

5.4 CoE Implementation and Case Study

In the previous sections, we have presented the CoE approach that aims to support and guide the automatic evolution of ontology-based context models at runtime. This section begins by describing the prototypical implementation of the proposed approach and then presenting a case study with three scenarios.

5.4.1 CoE Implementation

The proposed CoE approach was designed and implemented to address the problem of evolving users' initial ontology-based context models regarding arisen changes in the users' surrounding environments at runtime. This approach has been designed as multi-module architecture, as illustrated in Figure 5.1, with 4 modules: data source selection, data source unification, ontology-based context model learning and ontology-based context model integration. The whole architecture implementation will be described, starting from the data source selection module. This architecture consists of two different kinds of layers, backend and frontend. From the figure, we can see that our approach uses Spring Boot used as backend framework deployed on an embedded Tomcat server, while Angular is used as the primary frontend framework for building Web applications. The basic reason behind the Spring Boot choice is the availability of classes for ontology and XML manipulation, since it is an open-source Java-based framework, and the simple development of microservices based RESTful APIs. Therefore, both backend and frontend were implemented independently with only RESTful Web services as a bridge in between since all CoE's modules in the backend were implemented as RESTful Web services using Java technologies.

To accomplish the backend layer implementation:

1. ***The data source selection module.*** is where the process of search engine and information retrieval begins. Implementation-wise, we use Apache Lucene [Lucene, 2010] as a tool for information retrieval through data in order to implement this module. Prior to the discussion of the use of Lucene into our proposed search engine for selecting the most relevant semi-structured data source that could answer the context changes, it is beneficial to define the tool itself in clear as well as justification of why such an endeavor is worthwhile. Apache Lucene is the most popular and well-supported information retrieval tool that provides Java-based indexing and search technology. We selected Apache Lucene because it is popular, well-known, well-supported, actively developed, well-documented, and has a robust Java API. In addition, it provides facilities for pre-processing documents, to index the documents and to perform the keyword-based search. The implemented Lucene performs the following tasks: (i) the indexing of the semi-structured data sources entries in the data source repository; (ii) the querying to find the list of appropriate data sources that are similar to the captured context change keywords and (iii) the ranking of the retrieved list to select the most relevant one.

2. ***The data source unification module.*** takes as input the candidate data source and performs the first transformation to generate the corresponding XML pivot model by applying mapping rules defined and detailed above in Table 5.1. Prior to applying these mapping rules, they were written in Java to transform the candidate semi-structured data source to XML data.

3. ***The ontology-based context learning module.*** first applies the second transformation. For this, several freely-available APIs for java, namely: XML-Schema Object Model (XSOM) parser and Jena framework are used. XSOM was utilized to parse the generated XML schema and inspect the elements and attributes in it. In addition, Jena is used to generate the local ontology and fill it via concepts, relations, properties, and individuals according to the mapping rules defined in Table 5.2. Then, for the refinement phase, the extended Java WordNet Library (extJWNL), DBpedia were used together with the LSP, Hearst's and Cea's patterns described in the previous section. These patterns were implemented using Java. Moreover, to obtain the syntactical structure for each sentence in a metadata, we have implemented a Java process using CoreNLP Library. This process applies on a given metadata the following NLP techniques: sentence splitter, tokenizer, lemmatizer, POS tagger and dependency parser.

4. ***The ontology-based context model integration module.*** first implements both syntactic and semantic similarity measures as in Equation 5.1 to find a set of matches between the different entities extracted from the refined local ontology and the initial ontology-based context model. Second, it makes use of Jena for the second time to

enrich and populate the global ontology with concepts, relations, properties and individuals from the local ontology according to the set of matches.

5.4.2 CoE Case Study

In order to demonstrate the use of the implemented CoE approach in practice and to show how it behaves in realistic settings, this section presents a case study with three scenarios in HAR. These scenarios involve an elderly father, named Tom, and his young son, named Jean to illustrate different changes in their surrounding dynamic environments.

5.4.2.1 Scenario 1: "Moving from a hotel to an ordinary apartment"

In this first scenario, Jean stayed at a hotel for a couple of nights for a conference (t_0). The surrounding environment, in which he stays, contains sensors that are related to the hotel room, as shown in Figure 5.5(a). Obviously, Jean's initial ontology-based context model t_0 answered to his surrounding environment as depicted in Figure 5.6. Once the conference is completed, Jean checks out of the hotel to get his father's ordinary apartment (t_1). At t_1, the ordinary apartment, described in Figure 5.5(b), is visited by Jean. As illustrated in Figure 5.5(b), only sensors commonly found in their mobile devices are available in this ordinary apartment. Consequently, the initial ontology-based context model, previously presented in Figure 5.6, would answer to the new Jean's environment illustrated in Figure 5.5(b).

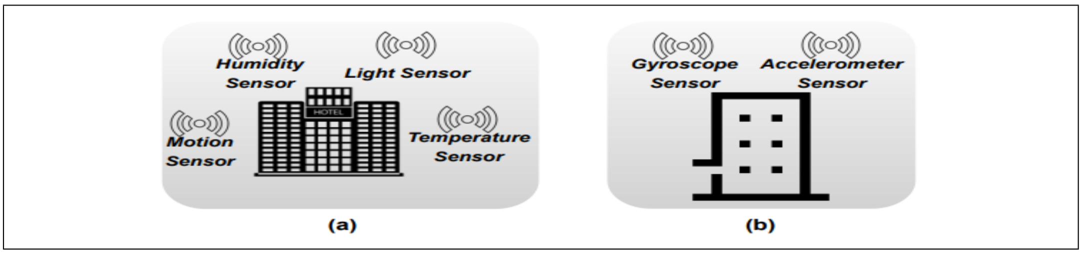

Figure 5.5: Jean's surrounding environments (a) at t_0 and (b) at t_1.

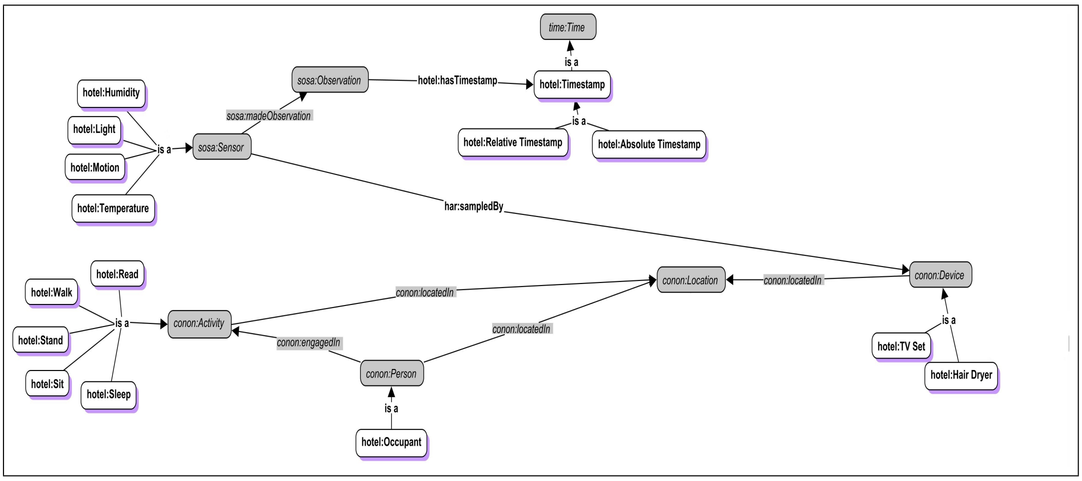

Figure 5.6: An excerpt of Jean's initial ontology-based context model at t_0.

To do so, the data source selection applies the automated data source search engine to select the appropriate data source using Apache Lucene. Figure 5.7 shows that the Heterogeneity Human Activity Recognition (HHAR) data source [Stisen et al., 2015] is selected for the evolution of Jean's model at t_1. The candidate data source includes accelerometer and gyroscope data recorded from smartphones and smartwatches in CSV format. It is accompanied by a metadata, a sample fragment of which is depicted in Figure 5.8.

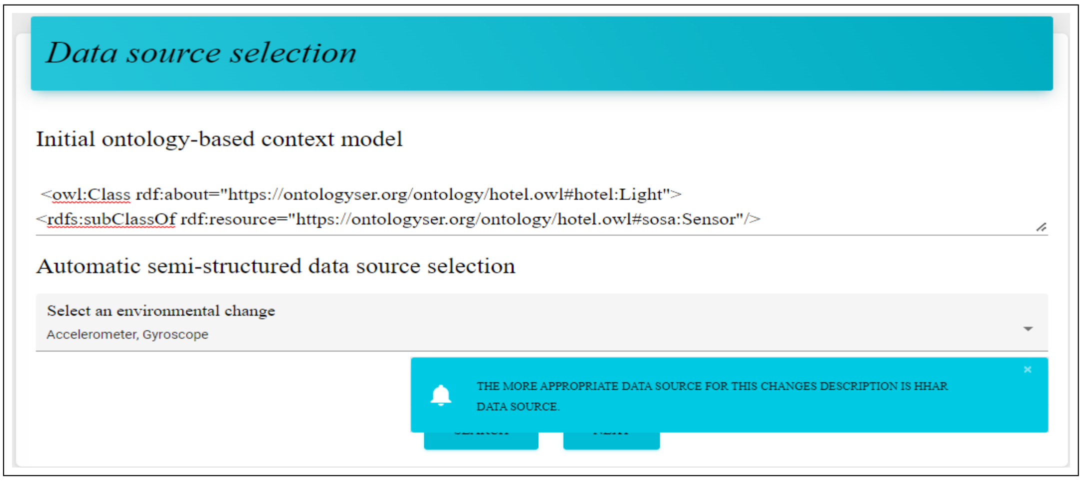

Figure 5.7: Candidate data source selection at t_1, the HHAR data source.

```
------------- Description ------------
User has Smatphone.
Sensor is sampled by Smartphone.
Sensor such as Accelerometer and Gyroscope.
Attributes of Accelerometer are 'acc_X' and 'acc_Y' and 'acc_Z'.
Activity is performed by User.
Activity such as sit, stand, walk, upstairs and downstairs.
Time has Arrival_Time.
Time has Creation_Time.
```

Figure 5.8: A sample fragment of the HHAR metadata.

Next, the data source format unification triggers the generation of a hierarchical XML pivot model from the HHAR data source. It starts by parsing the candidate data source after performing the preprocessing and preparation step to extract its schema information as observed in Figure 5.9. Once this step is completed, the different mapping rules previously described in Table 5.1 are explored to transform the parsed HHAR data source to XML data. The output is a hierarchical XML pivot model including an XML document together with an XML schema document, whose results are depicted in Figure 5.10.

Then, the previously generated XML schema was traversed to parse its elements and attributes using the XSOM parser, which allows to extract the schema of the XML pivot model and inspect information into it. The output of XSOM is represented in Figure 5.11.

After that, the output of XSOM is used as input together with the set of mapping rules defined in Table 5.2 in order to generate the local ontology that corresponds to the HHAR data source. An excerpt of this local ontology is shown in Figure 5.12. For example, the two complex type elements "stand" and "sit" are mapped to ontology sub-concepts, whose super concept is the simple type element "SuperElem_Activity". Moreover, in this figure, we

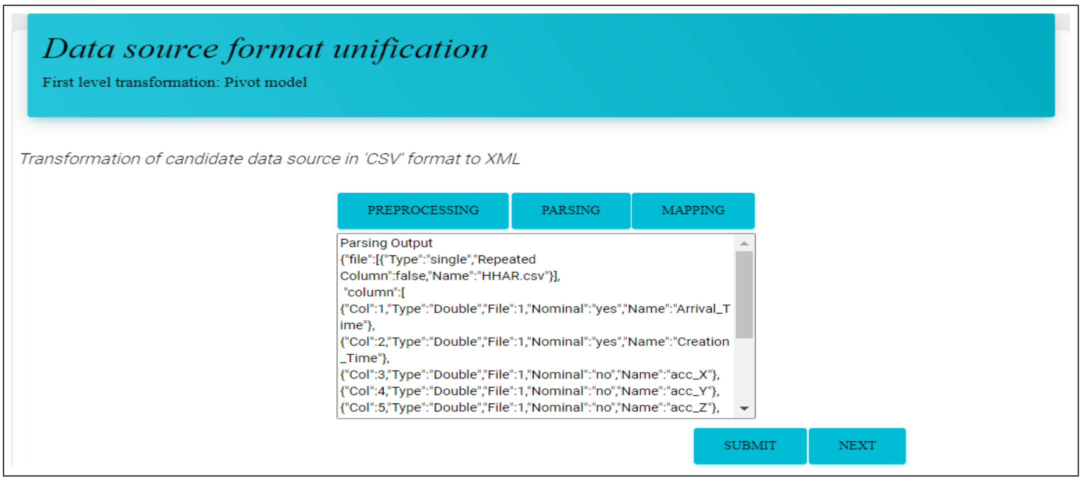

Figure 5.9: Data source format unification at t_1 with an example of a parsed HHAR data source.

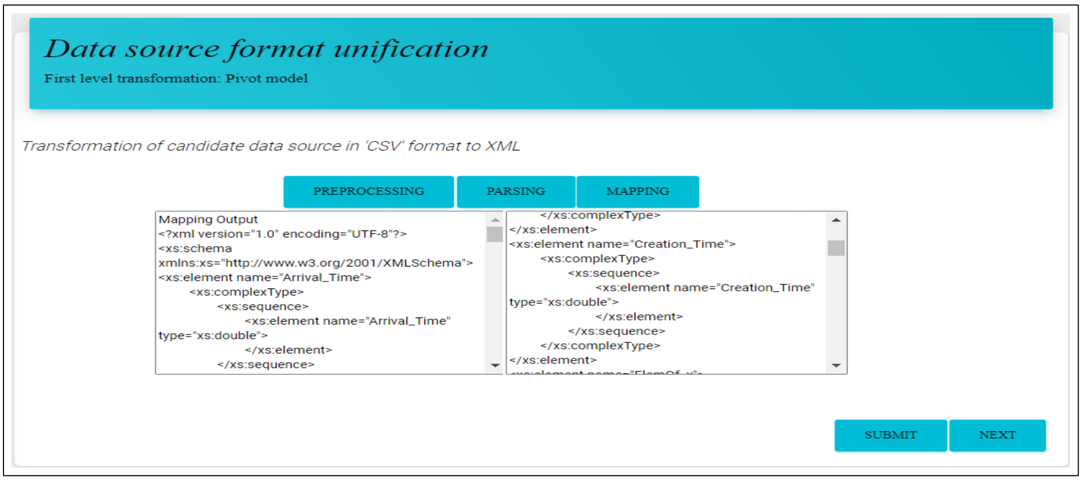

Figure 5.10: Data source format unification at t_1 with an example of XSD.

can notice that the generated local ontology contains some unrelated concepts. Therefore, some refinements are required to discover missing non-taxonomic relations.

Following this, the third module uses the generated local ontology together with the set of background knowledge to handle some refinements. In this case, for example, a concept refinement is performed where, the LSPs for missing concept acquisition, earlier presented in Table 5.3, are used to recognize sentences included in the HHAR metadata and to match them with such LSP for identifying missing concepts as shown in Figure 5.13. As seen in this figure, the sentence "Attributes of Accelerometer are 'acc_X' and 'acc_Y' and 'acc_Z'." could match with LSP "Property|ies | characteristic|s | attribute|s of NP<class> be [PARA] [(NP<property>,) * and] NP<property>", and a new concept "Accelerometer" is identified as the domain of "acc_X", "acc_Y" and "acc_Z". Consequently, the "hhar:ElemOf_x" concept was replaced by the "hhar:Accelerometer" concept. The output of the third module is the refined local ontology, an excerpt of which is shown in Figure 5.14. By accomplishing refinement, the last module updates Jean's initial ontology-based context model. As a result, at t_1, an evolved ontology-based context model is generated, an excerpt of which is depicted in Figure 5.15.

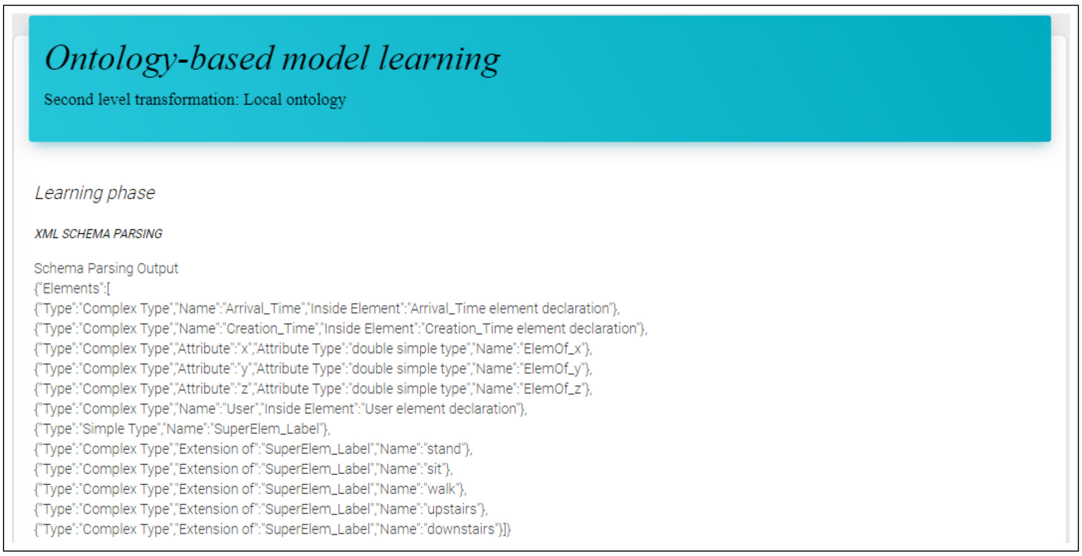

Figure 5.11: Ontology-based model learning at t_1 in the learning phase, an example of a parsed XSD.

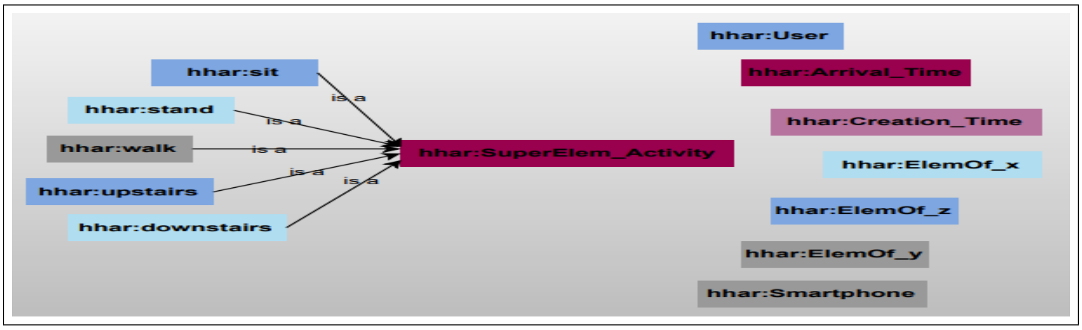

Figure 5.12: Ontology-based model learning at t_1 in the learning phase, an excerpt of the local ontology.

5.4.2.2 Scenario 2: "Moving from an ordinary apartment to smart apartments"

In this second scenario, Jean lived with his father in an ordinary apartment during the years of study (t_1). The surrounding environment in which they live contains only sensors commonly found in their mobile devices, as shown in Figure 5.16(a). Obviously, Tom and Jean's initial ontology-based context model t_1 answered to their surrounding environment as depicted in Figure 5.17. Then, after Jean graduated from the university, Tom and Jean decided to move to two twin smart apartments in the same building that are equipped with sensors. At t_2, this new environment, described in Figure 5.16(b), is visited by Tom and Jean. As illustrated in Figure 5.16(b), a range of sensors from magnetic to electric, flush, PIR, and pressure sensors are available in the new Tom and Jean's smart apartments. Consequently, the initial ontology-based context model, previously presented in Figure 5.17, would answer to the new Tom and Jean's environment illustrated in Figure 5.16(b).

To provide a strong answer, the data source selection applies the automated data source search engine to select the candidate data source. Figure 5.18 shows that it adopts the Ordonez data source [Ordónez et al., 2013] for the evolution of Tom and Jean's model at t_2. Ordonez data source contained 20,358 observations in CSV format conducted with

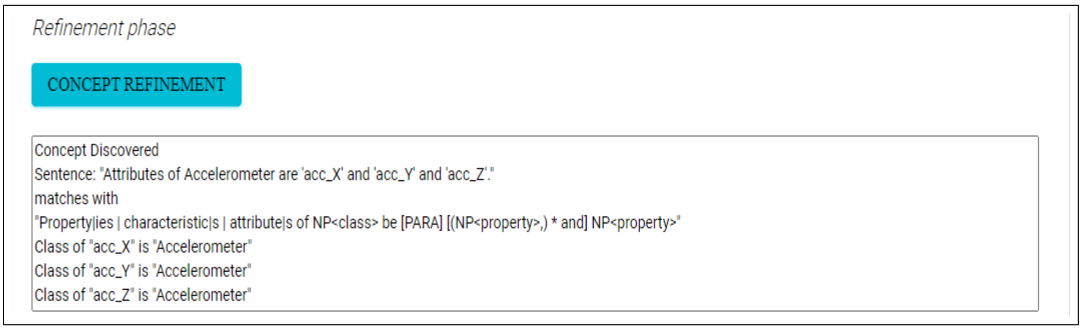

Figure 5.13: Ontology-based model learning at t_1 in the refinement phase. An example of concept refinement using LSPs.

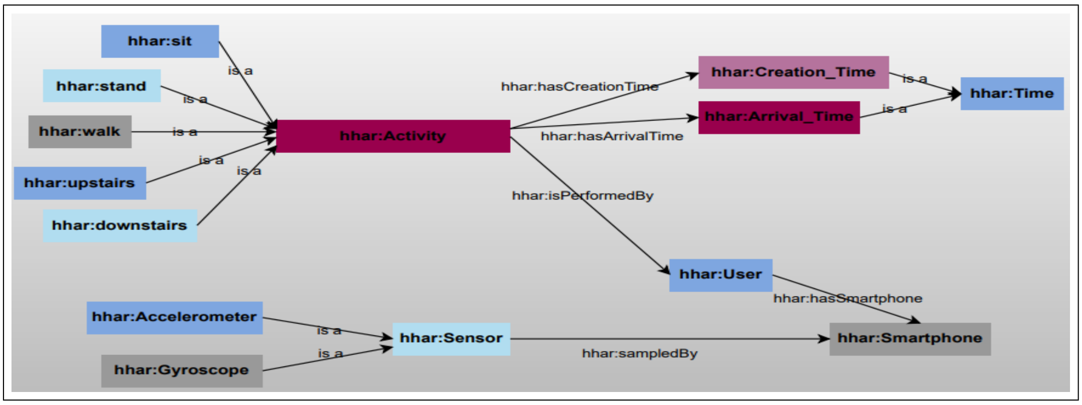

Figure 5.14: Ontology-based model learning at t_1 in the refinement phase, an excerpt of the refined local ontology.

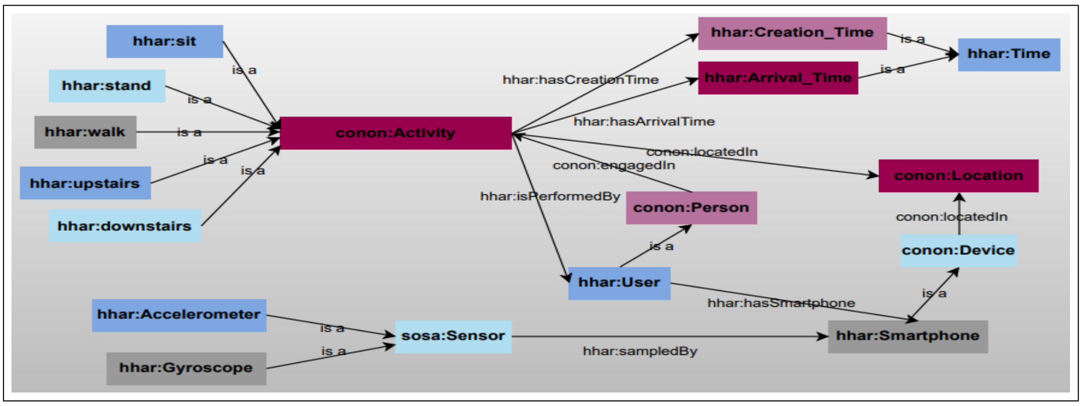

Figure 5.15: Ontology-based model evolution at t_1, showing an excerpt of Jean's evolved ontology-based context model at t_1.

14 sensors. These sensors captured about 10 basic activities. The Ordonez data source is accompanied by metadata, a sample fragment of which is depicted in Figure 5.19.

Next, the second and third modules use the Ordonez data source together with the set of background knowledge to learn and refine a local ontology. The output of the third module is the refined local ontology, an excerpt of which is shown in Figure 5.20.

After the refinement, the last module updates Tom and Jean's initial ontology-based context model. As a result, at t_2, an evolved ontology-based context model is generated,

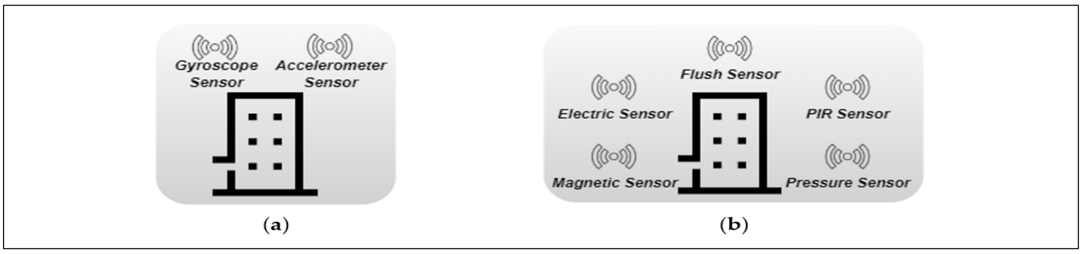

Figure 5.16: Tom and Jean's surrounding environments (a) at t_1 and (b) at t_2.

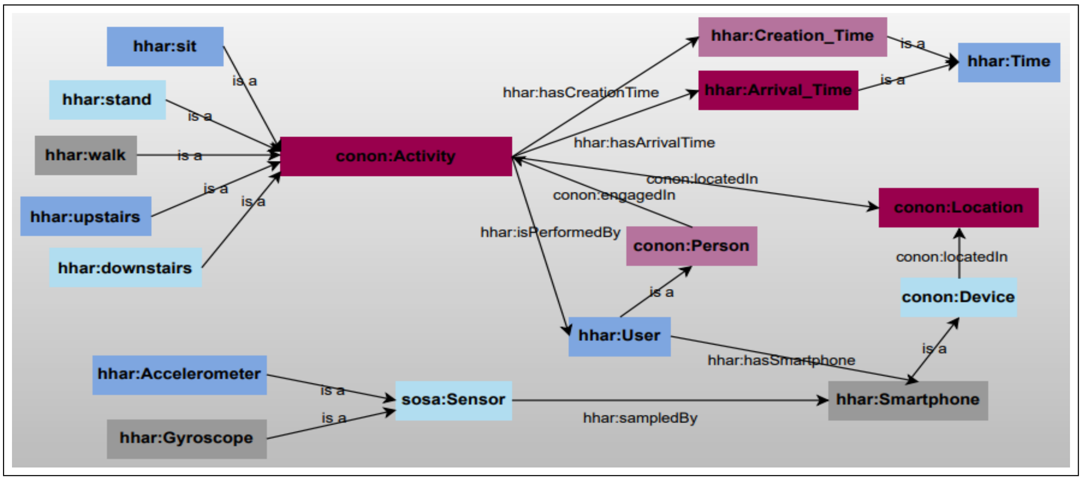

Figure 5.17: An excerpt of Tom and Jean's initial ontology-based context model at t_1.

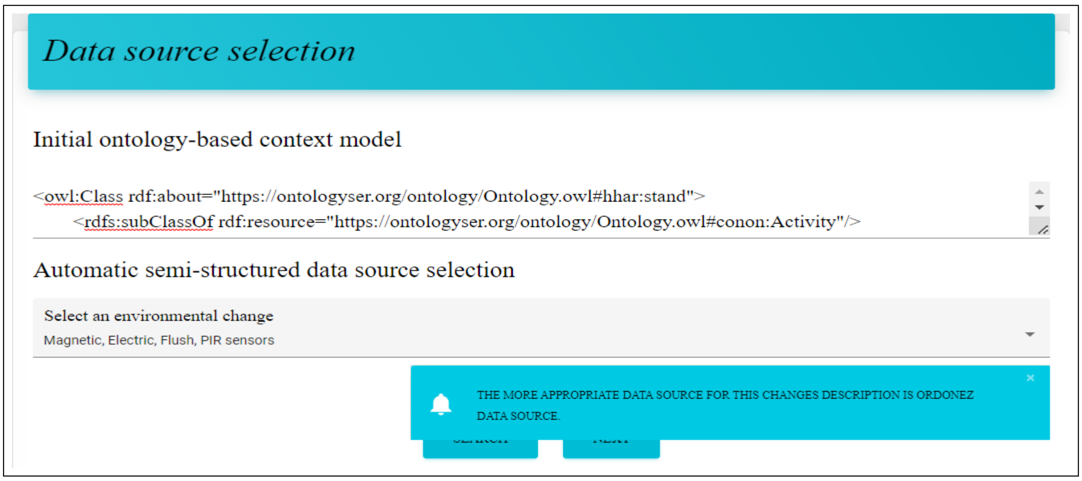

Figure 5.18: Candidate data source selection at t_2, the Ordonez data source.

an excerpt of which is depicted in Figure 5.21.

5.4.2.3 Scenario 3: "Moving from a smart apartment to a smart home"

In this third scenario, Tom and Jean lived in their smart apartments represented in Figure 5.22(a), at t_2. Obviously, Tom and Jean's initial ontology-based context model answered to their surrounding environment at t_2 as depicted in Figure 5.21. Then, after a period of time, Tom and Jean move out of their smart apartments and into an independent smart

```
Ordonez  DATASET DESCRIPTION
============================
This dataset comprises information regarding the ADLs performed by two users on a daily basis
in their own apartments.
Smart apartment setting: 5 rooms
Activities (ADLs included): Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner,
Snack, Spare_Time/TV, Grooming
Sensors:        PIR: Shower, Basin, Door Kitchen, Door Bathroom, Door Bedroom
                Magnetic: Maindoor, Fridge, Cupboard
                Flush: Toilet
                Pressure: Seat, Bed
                Electric: Microwave

SENSOR EXPLANATION
Sensor such as Magnetic, Flush, Pressure, Electric and PIR has place in the apartment.

ACTIVITY EXPLANATION
Leaving, toileting, showering, sleeping, breakfast, lunch, dinner, snack, spare_Time/TV
and grooming belong to activity.

These activities held in different places.
```

Figure 5.19: A sample fragment of the Ordonez metadata.

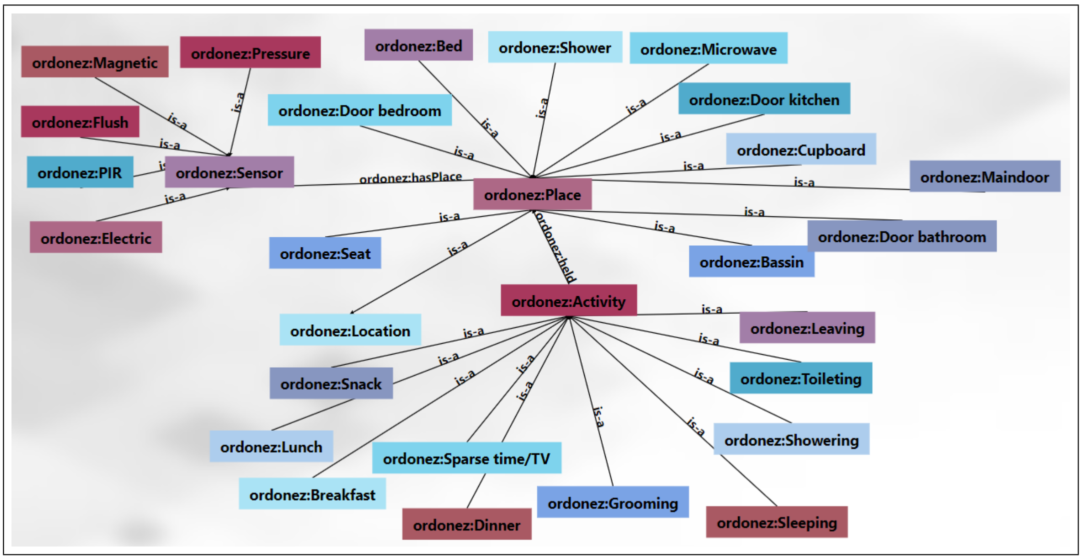

Figure 5.20: Ontology-based model learning at t_2, the refinement phase, showing an excerpt of the refined local ontology.

home with more advanced sensors. At t_3, this new and different pervasive environment, described in Figure 5.22(b), is visited by Tom and Jean. As illustrated in Figure 5.22(b), a range of sensors including distance, sonar, force, temperature, photocell, and contact to infrared (IR) are available in Tom and Jean's new smart home. Consequently, the initial ontology-based context model, previously presented in Figure 5.21, would answer to the new Tom and Jean's pervasive environment illustrated in Figure 5.22(b).

To provide a strong answer, the data source selection queries the data source search engine to select the appropriate data source. Figure 5.23 shows that ARAS data source [Alemdar et al., 2013] is selected for the evolution of Tom and Jean's model at t_3, since this data source had a stronger tendency for having the same features of the new smart home. The ARAS data source is well-known in the literature for having collected and published a home automation data source for daily living in smart homes. It contains 5,184,000 observations in CSV format conducted with 20 sensors. These sensors captured

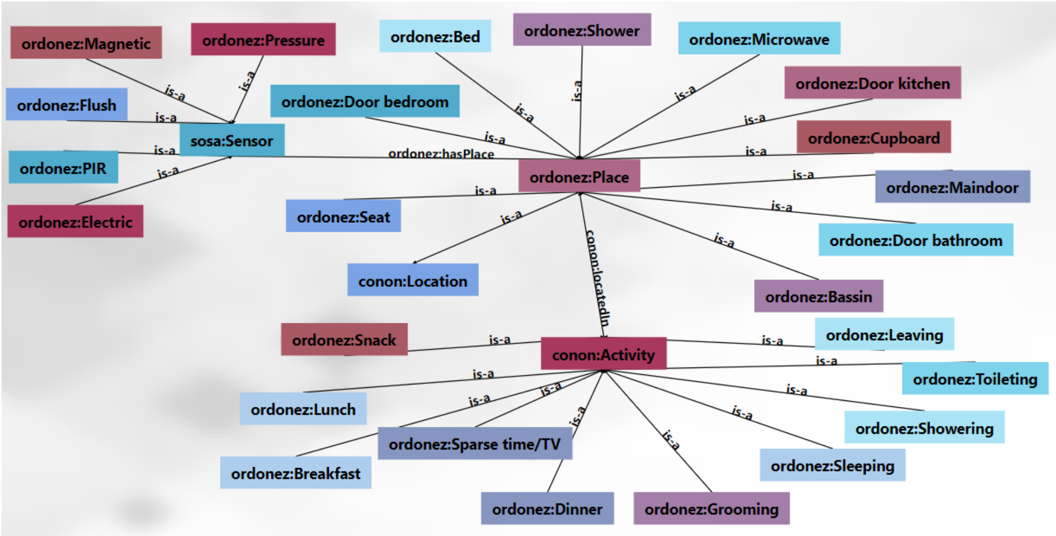

Figure 5.21: Ontology-based model evolution at t_2, showing an excerpt of Tom and Jean's evolved ontology-based context model at t_2.

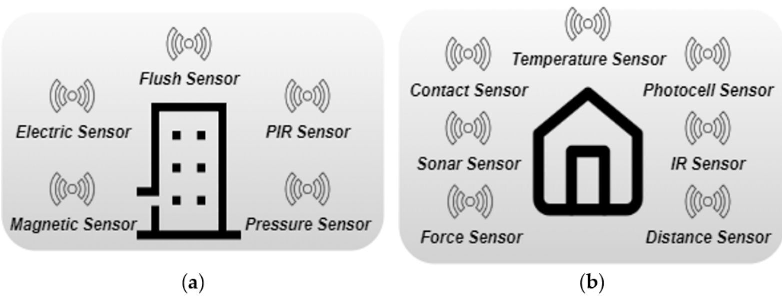

Figure 5.22: Tom and Jean's surrounding environments (a) at t_2 and (b) at t_3.

about 27 different activities. It is accompanied by metadata, a sample fragment of which is depicted in Figure 5.24.

Next, the data source format unification for the generation of a hierarchical XML pivot model from the ARAS data source, is performed. It begins with parsing this candidate data source after performing the preprocessing and preparation step to extract its schema information as depicted in Figure 5.25. Then, the set of mapping rules previously described in Table 5.1 is explored to transform the parsed candidate data source to a hierarchical XML pivot model, whose results are represented in Figure 5.26.

Then, the pivot model is traversed to parse its elements and attributes using the XSOM parser as shown in Figure 5.27. For instance, the "SuperElem_Activity" feature is defined as a simple type, while the "Going Out" and "Having Shower" features are defined as a complex type with a complex content extension.

After that, the third module uses the XSOM output as an input together with the set of mapping rules defined in Table 5.2 to generate a local ontology model using the Jena API. The second output of the third process is the resulting local ontology, where an excerpt of the local ontology is shown in Figure 5.28. Thus, for example, the two complex type elements "Going Out" and "Having Shower" are mapped to ontology sub-concepts, whose

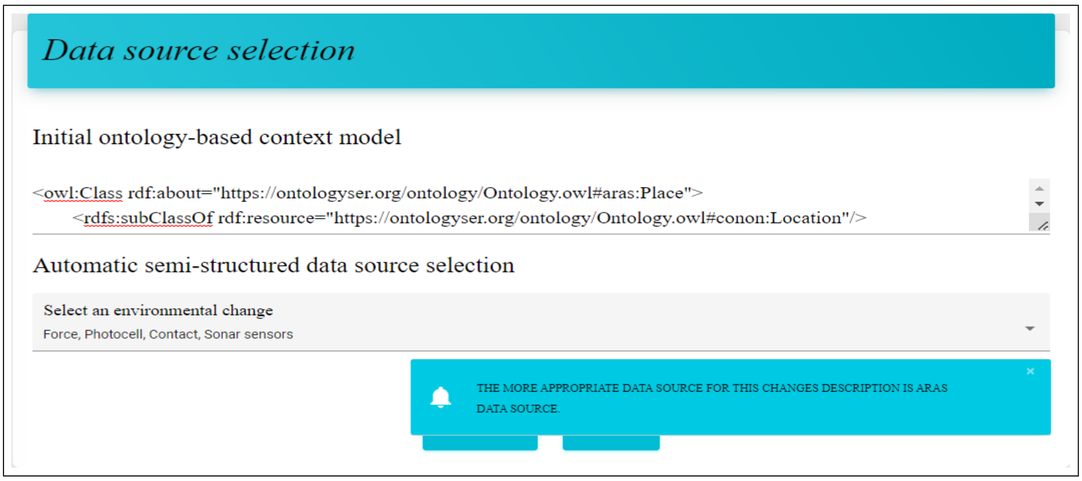

Figure 5.23: Candidate data source selection at t_3 of the ARAS data source.

```
SENSOR EXPLANATION
There are several kinds of sensors: Photocell, Contact Sensor, Distance, Force
Sensor, Sonar Distance, Temperature and IR.

Each sensor has its place.

PLACE EXPLANATION
Places including Wardrobe, Couch, TV receiver, Chair, Fridge, Kitchen Drawer,
Bathroom Cabinet, House Door, Bathroom Door, Shower Cabinet Door, Hall, Kitchen,
Tap, Water Closet and Bed.

ACTIVITY EXPLANATION
Activity such as Changing Clothes, Having Guest, Having Conversation, Cleaning,
Listening to Music, Talking on the Phone, Brushing Teeth, Shaving, Laundry, Reading
Book, Using Internet, Napping, Toileting, Having Shower, Studying, Watching TV,
Sleeping, Having Snack, Washing Dishes, Having Dinner, Preparing Dinner, Having
Lunch, Preparing Lunch,Having Breakfast, Preparing Breakfast, Going Out, Other,
are included in ARAS dataset.

These activities performed at specific location such as home.
```

Figure 5.24: A sample fragment of the ARAS metadata.

super concept is the “SuperElem_Activity” simple type element.

Afterwards, this module allows for refining the local ontology and generating new elements if necessary. In this case, a taxonomic refinement is performed where, internally, the Hearst and Cea patterns and Matcher Java regex classes are explored to recognize sentences that are included in the ARAS metadata and to match them with such patterns for identifying missing taxonomic relations as shown in Figure 5.29(a). For instance, the second sentence matches well with Hearst’s pattern “NP , such as NP, * and|or NP”, and a new taxonomic relation is identified between the “Going Out” concept and “Activity” concept, since “Activity” is a hypernym of “Going Out”. Consequently, the “aras:SuperElem_Activity” is replaced by the “aras:Activity” concept. Additionally, extJWNL is exploited to accomplish the taxonomic refinement as shown in Figure 5.29(b). Thus, a new taxonomic relation is suggested between the “aras:Place” concept and “aras:Location” concept. Therefore, a definition of the “aras:Location” concept is created, and the suggested relation is built. Figure 5.30 presents an excerpt of the partially refined local ontology where all the taxonomic refinements are illustrated. In addition, non-taxonomic relation refinement is accomplished, since the hidden connections between all the learned super concepts were missing. In this case, for example, the metadata is looped to extract all the sentences that contain the label of the “aras:Sensor” concept. Then, analysis of the structure and dependencies

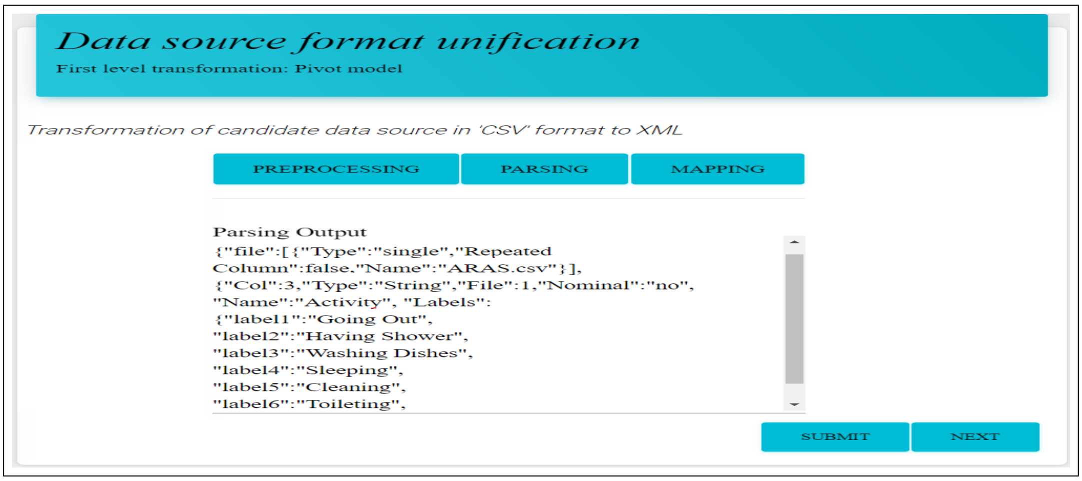

Figure 5.25: Data source format unification at t_3 with an example of a parsed candidate data source.

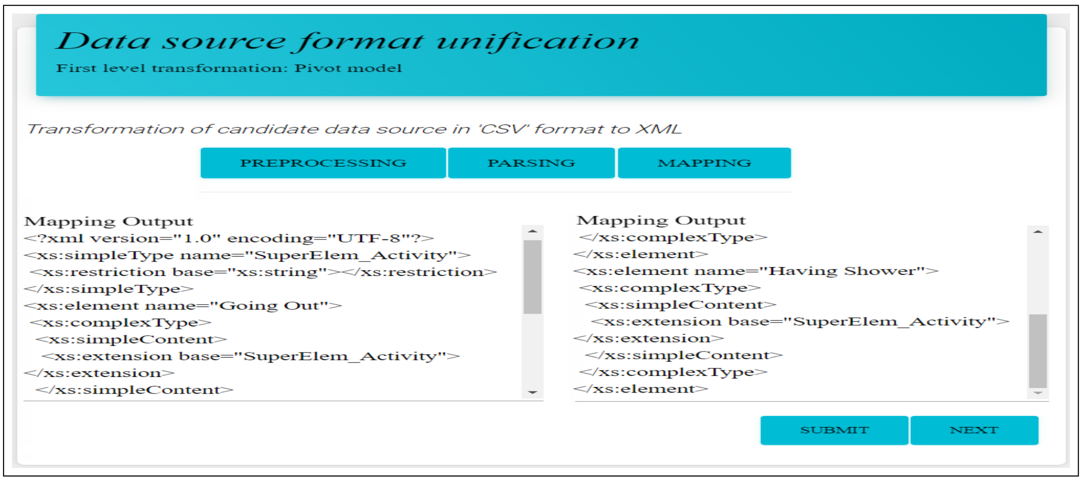

Figure 5.26: Data source format unification at t_3 with an example of XSD.

of the candidate sentence is performed, and as illustrated in Figure 5.31, the <"Sensor", "has", "Place"> triplet is extracted, where the "has" verb is an indicator for a non-taxonomic relation and used to label the new non-taxonomic relation between the "aras:Sensor" and "aras:Place" concepts. Figure 5.32 shows an excerpt of the refined local ontology, where all the non-taxonomic refinements are illustrated.

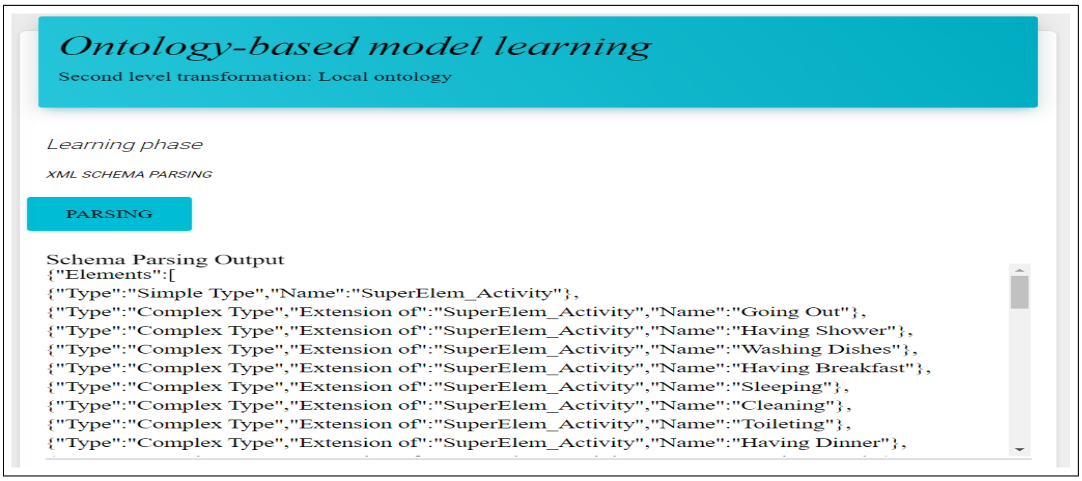

Figure 5.27: Ontology-based model learning at t_3 in the learning phase, an example of a parsed XSD.

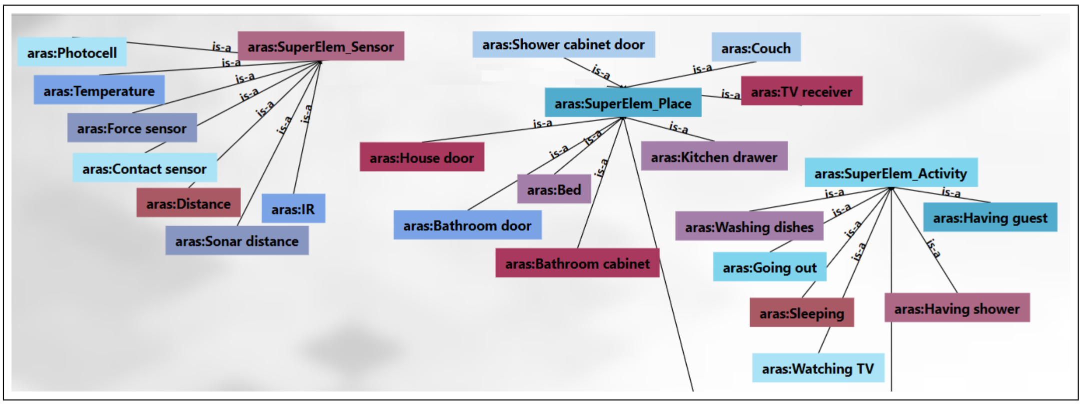

Figure 5.28: Ontology-based model learning at t_3 in the learning phase, an excerpt of the local ontology.

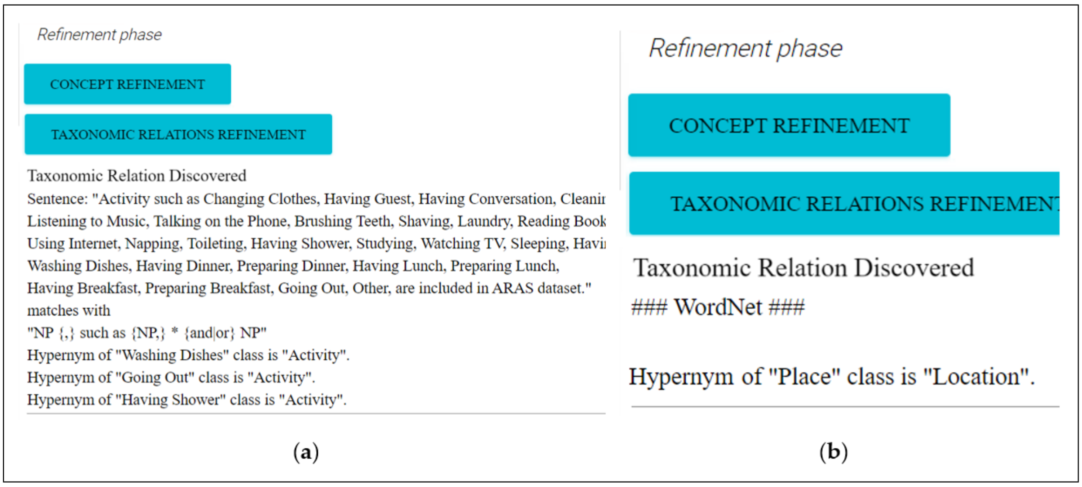

Figure 5.29: Ontology-based model learning at t_3 in the refinement phase. (a) An example of taxonomic refinement using LSPs. (b) An example of taxonomic refinement using WordNet.

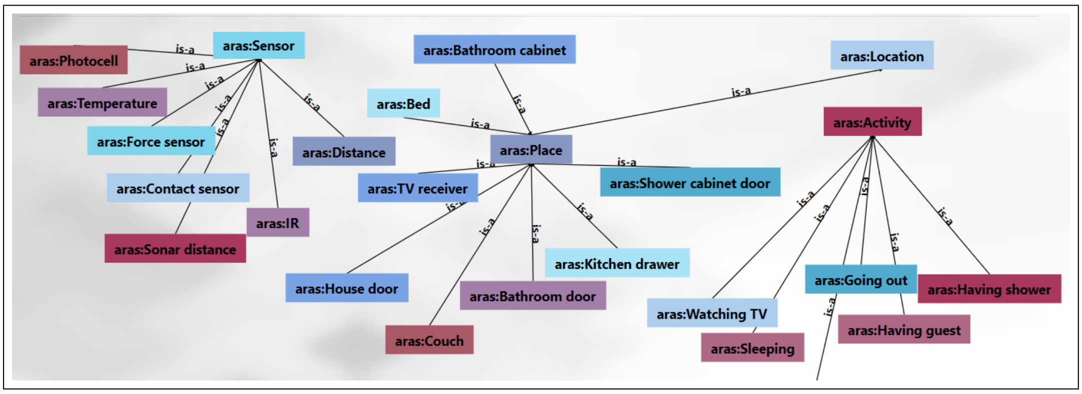

Figure 5.30: Ontology-based model learning at t_3 in the refinement phase, an excerpt of a partially refined local ontology after taxonomic refinements.

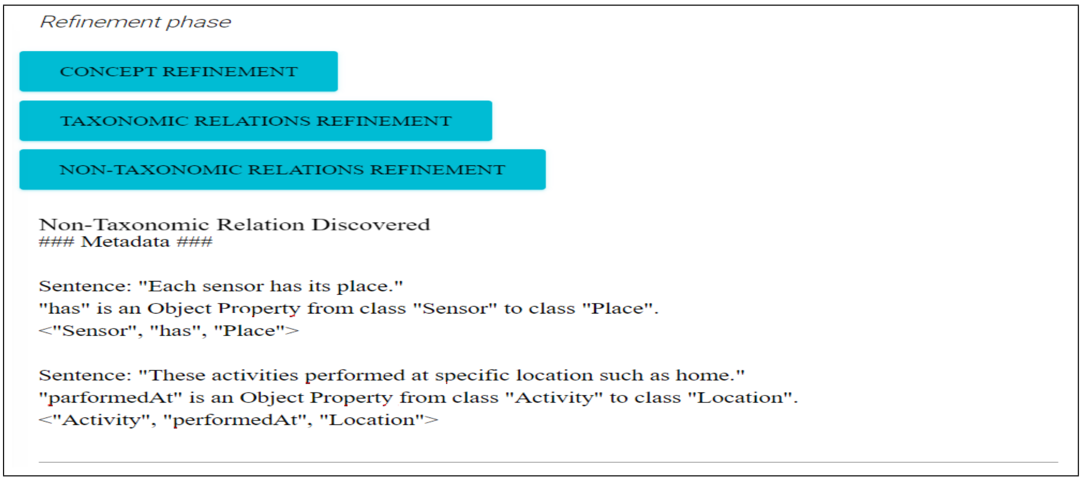

Figure 5.31: Ontology-based model learning at t_3 in the refinement phase, an example of non-taxonomic refinements.

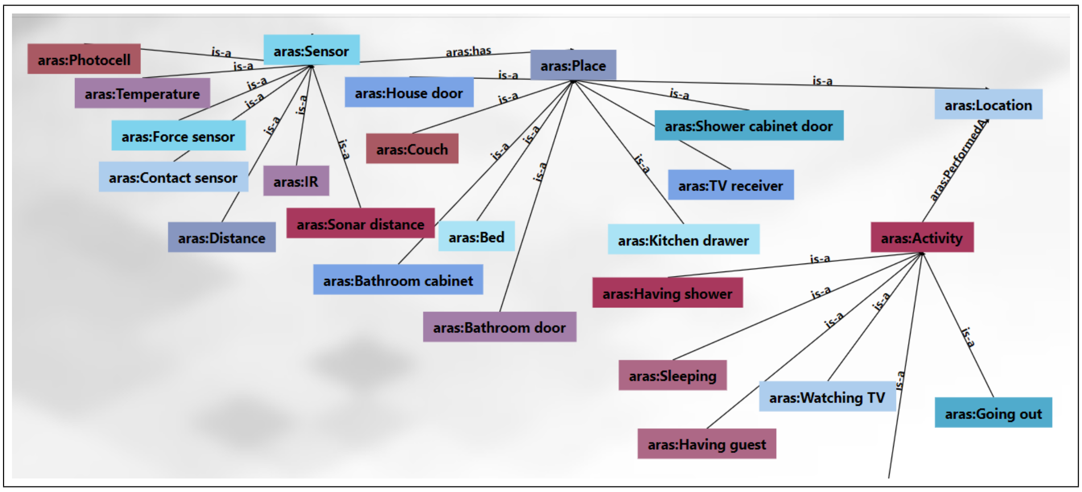

Figure 5.32: Ontology-based model learning at t_3 in the refinement phase, an excerpt of a partially refined local ontology after non-taxonomic refinements.

With the result of the refinement, the last module undertakes the evolution of Tom and Jean's initial ontology-based context model. For this, the syntactic and semantic similarities are applied to find out the similarities between the ontology terms during the alignment activity. Figure 5.33 shows a part of the alignment findings. For example, the "aras:Location", "aras:Sensor", and "aras:Activity" concepts are equivalent to the "conon:Location", "sosa:Sensor", and "conon:Activity" concepts, respectively. Finally, Tom and Jean's initial ontology-based context model is evolved based on the alignment results. The output of this module is Tom and Jean's evolved ontology-based context model at t_3, an excerpt of which is depicted in Figure 5.34. For instance, in the case of equivalent concepts with similar labels, such as "aras:Activity" and "conon:Activity", the original concept "conon:Activity" is kept, while in the case of dissimilar classes, such as "ordonez:Pressure" and "aras:Temperature", the new concept "aras:Temperature" is copied, and the old one, "ordonez:Pressure", is overlooked. In addition, for the case of relations, the "aras:PerformedAt" relation is neglected, since there existed a relation "conon:locatedIn" that held between the concept "conon:Activity" and the concept "conon:Location". By contrast, the new relation "aras:hasPlace" is copied into the evolved ontology-based context model to relate between the old concept "sosa:Sensor" and the new concept "aras:Place".

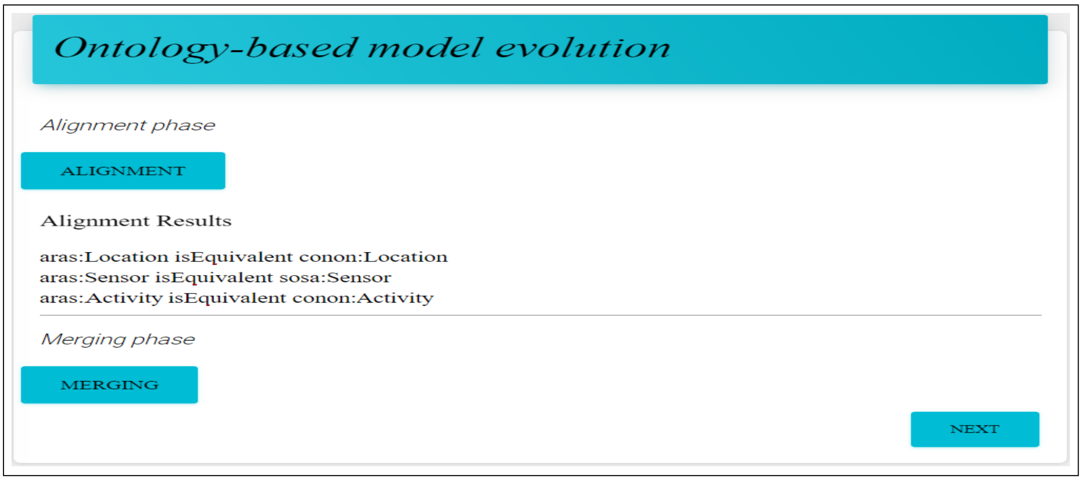

Figure 5.33: Ontology-based model evolution at t_3 in the alignment phase, an example of the main alignment findings.

5.5 Concluding Remarks

In this chapter, we have explained the first approach in the IConAS approach to evolving ontology-based context models in an automatic way. The CoE is the first approach into this path. This approach is based on an unsupervised ontology learning approach that is capable of enriching and populating an existing context model from a semi-structured data source. It was designed to be general enough to be applicable to any format of semi-structured data sources, thus, not being oriented towards one specific format, thanks to the two-level transformations. In summary, this CoE approach introduced for evolving existing ontology-based context models according to changes occurring in dynamic surroundings at runtime. Therefore, this chapter has presented a detailed illustration of the context evolution approach as follows:

- Firstly, in section 5.2, we have provided an overview of the CoE approach.

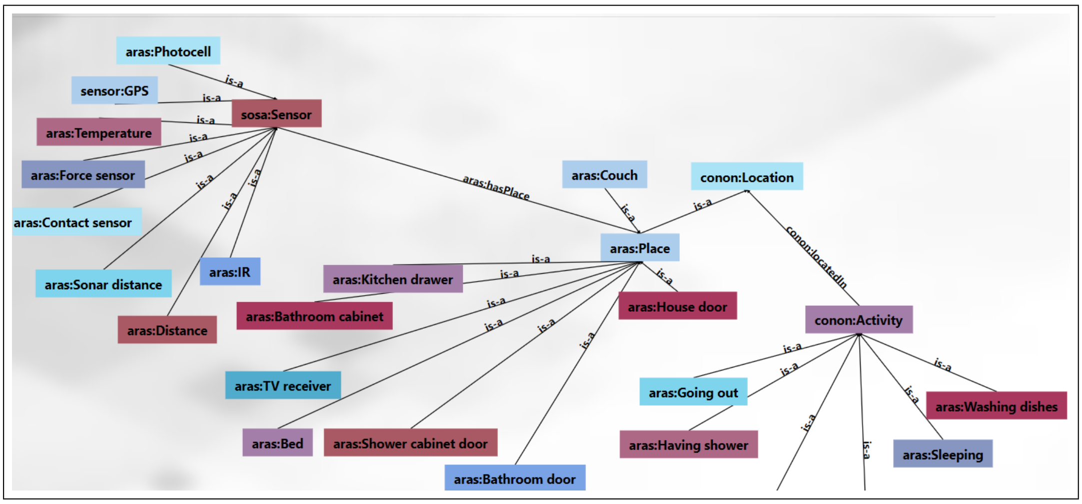

Figure 5.34: Ontology-based model evolution at t_3, showing an excerpt of Tom and Jean's evolved ontology-based context model at t_3.

- Secondly, in section 5.3, we have introduced the general architecture of the CoE approach. Then, we have described the key modules defined in this architecture.
- Finally, in section 5.4, we have first proposed an implementation of the CoE approach. Then, we have described a case study with different scenarios.

Further evaluations of CoE approach will be conducted in chapter 7. They will be presented in Section 7.2.

CHAPTER 6

Decision-Making Adaptation Approach

6.1 Introduction

One of the most expensive activities in a context-aware solution is the decision-making adaptation at runtime. As revealed by chapter 1 and chapter 3, this problem might be more severe in dynamic environments, where the surrounding environment and context may change at runtime. One of possible ways of reducing this cost is the decision-making adaptation at runtime. Unfortunately, dynamic decision-making adaptation at runtime is not as straightforward. There is a considerable challenge for any dynamic decision-making adaptation approach to enrich rule knowledge bases. In this chapter, we introduce the (DMA) approach, which forms the first part of the IConAS approach, to address this problem of effectiveness in dynamic adaptation of decision-making at runtime. For that, we target the second sub-objective:

- <O.2>: to provide another approach to automatically adjust behaviors of context-aware solutions in the wake of changes entailed in their surroundings as well as their context models at runtime, where we investigate the following research questions:
  - RQ.2. How can a decision-making process be adapted in an automatic manner at runtime?
    - RQ.2.1. How can we automatically enrich rule knowledge bases with missing decision rules and keep them up-to-date?

The rest of this chapter is organized as follows. Section 6.2 gives a general overview of the proposed DMA approach. Section 6.3 presents the DMA architecture in discussing the decision-making adaptation modules' details for enriching a rule knowledge base to cope with context changes at runtime. Section 6.4 presents implementation details and case studies to demonstrate how the implementation can be utilized to automatically learn and generate new decision rules in order to enrich existing rule knowledge bases at runtime. Section 6.5 ends this chapter with a few concluding remarks.

6.2 DMA Approach Overview

In context-aware computing, there exists a shift from static environments to dynamic environments [Da Rocha and Endler, 2012]. This shift reflects a growing interest devoted to dynamic environments. With the growing interest, a crucial need is revealed for context-aware solutions to be aware of and to adapt to their changing contexts in highly dynamic environments at runtime [Chang, 2016]. To support this need, a grand challenge is that context-aware solutions should adjust their behaviors to the dynamics entailed in their surrounding environments at runtime. In order to meet this challenge, a decision-making process improvement by providing appropriate decisions to users situated in highly changing environments, has emerged to make context-aware solutions more resilient to dynamic environments at runtime.

To achieve this sustained need, we propose the DMA approach [Jabla et al., 2022a, Jabla et al., 2021b] that aims to automatically improve decision-making process to support dynamic environments at runtime. The main feature of the proposed approach is to offer a rule knowledge base, where decision rules are fluid and evolutive at runtime for alleviating the burden of manually creating rules to react towards users' preferences and behaviors changes or environment changes. The novelty and contribution of this approach could be drawn from three-fold:

- First, we present a novel hybrid learning approach towards effectively generating a concise set of non-redundant rules in automated fashion by applying two Machine Learning algorithms. We hybridize Machine Learning algorithms to generate a more accurate and complete set of rules, to avoid redundancy and to build a strong rule knowledge base in a context-aware solution, in which we are interested;

- Second, we extend a Genetic Algorithm (GA) [Goldberg, 1989] with a multi-analysis technique in the direction of rule optimization. The rule optimization is applied to find the well-performed rules from the generated rules since Machine Learning algorithms are not much proficient at optimizing rules;

- Third, we introduce an automatic transformation of obtained decision rules to rules expressed in Jena [Carroll et al., 2004]. Rule transformation is performed to express the well-performed rules in such a way that they can run over an ontology-based context model and a context-aware solution can reason.

6.3 DMA Approach Architecture

For supporting the improvement of decision-making in dynamic environments at runtime, the DMA approach typically consists of two main modules, namely, rule generation module and rule transformation module. Figure 6.1 illustrates a schematic view of the present approach architecture.

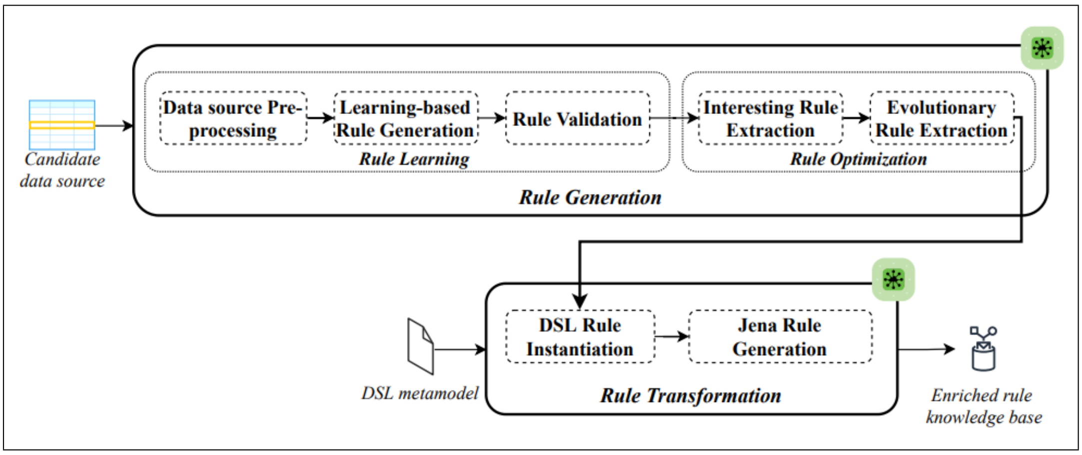

Figure 6.1: DMA approach architecture.

In the following Subections 6.3.1 and 6.3.2, we discuss these modules and their roles in generating decision rules and transforming them into Jena rules.

6.3.1 Rule Generation Module

The rule generation module is designed to automatically discover decision rules needed to meet the changes occurring in users' dynamic environments. The concept of decision rules is a useful method for discovering correlation and relationships between objects in a data source since it is based on statistical analysis and artificial intelligence. This method is particularly appropriate for analyzing the correlations between objects among input data sources and producing rules that are served as decision rules of the form IF-THEN. Therefore, the rule generation module aims to preprocess a candidate data source, to generate decision trees and to infer the well-performed decision rules resulting from decision rules mining in order to further enrich a rule knowledge base and improve decision-making process at runtime. The present module runs every time when new changes are arrived at runtime and a priori rules are deemed not relevant to these changes. Basically, this module is defined considering two essential phases as shown in Figure 6.1.

6.3.1.1 Rule Learning

The rule learning is the core phase where learning and extracting decision rules from a candidate data source are performed. In this sense, various steps starting from the pre-processing of the data source to the validation of rules are considered.

- **Data source pre-processing.**

  This represents the first step of the rule learning phase, which is applied to improve the quality of the candidate data source so as to ensure accurate, consistent and complete generated decision trees and further decision rules. This step is responsible for preparing data from the candidate data source before going to generating rules based on learning. Hence, pre-processing of data source is done in the following major stages:

  - **Data cleaning.** This performs both operations like filling missing data and smoothing noisy data. It aims to replace missing data with the average of existing data, using different traditional imputation methods such as mean, median, etc. To smooth noisy data, we carry out techniques, such as clustering, regression and binning, to eliminate the noise in the data source as it rises due to random variation;

  – **Data discretization.** This consists of transforming a continuous data into a categorical or nominal data. This stage is particularly required when the Machine Learning algorithm cannot cope with continuous data;

  – **Data reduction.** This is needed to have a simplified representation of a data source with relevant data that can work effectively with a Machine Learning algorithm. To do it in the simplest manner, data reduction stage removes redundant and inconsistent data in data sources;

- **Learning-based rule generation.**

  After the data pre-processing step is completed, the learning-based rule generation step takes place in the rule learning phase, which aims to first to automatically learn training models from the candidate data source for building decision trees using Machine Learning algorithms, then to discover hidden knowledge patterns in the form of decision rules and finally to integrate them into the temporary rule repository. To this end, the present step provides two elementary mechanisms that are simply described in Figure 6.2.

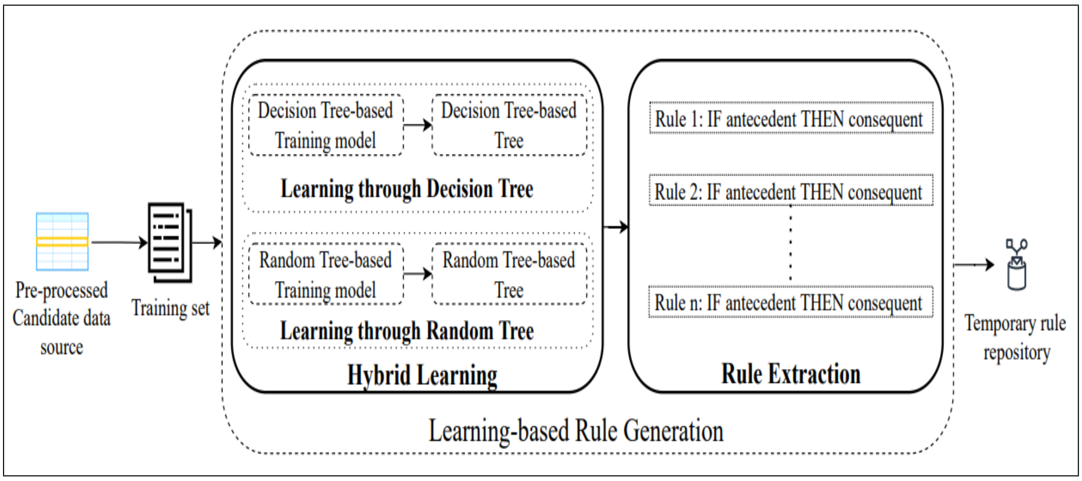

Figure 6.2: Learning-based rule generation step.

A first mechanism, called hybrid learning, takes advantage of Machine Learning and pattern recognition in learning about data, extracting relevant hidden knowledge patterns and eliminating redundancy. Thus, two widely adopted Machine Learning algorithms, namely J48 Decision Tree [Quinlan, 1986] and Random Tree [Witten and Frank, 2002] are combined to design a hybrid mechanism. We came up with the idea of hybridizing J48 Decision Tree and Random Tree algorithms to improve the accuracy of learning as they are tree-based techniques that provide the highest accuracy-diagram of a decision tree [Saad and Nagarur, 2020]. As far as we know, it is the first time that J48 Decision Tree and Random Tree algorithms are combined to rule learning. Both Machine Learning algorithms are applied to train the pre-processed data source and create the training models including decision trees. The resulting training models consist of a set of decisions in a tree structure, which could be utilized to generate rules from each leaf node [Suresh et al., 2020]. Algorithm 6.1 gives an outline on how to generate rules from the candidate data source with a hybrid mechanism. As shown in Algorithm 6.1, the details of how to generate the decision trees statements are not presented here. The algorithm takes a candidate data source as its argument and returns the decision trees discovered from the candidate data source. In line 2, we start to train the candidate data source over the J48 Decision Tree algorithm to build the corresponding decision tree on the decision tree-based training model. In line 3, we use the Random Tree algorithm to train the candidate data source in order to get the second decision tree on the random tree-based training model. In line 4, we extract the different decision trees from both obtained training models.

```
1 input: DS a pre-processed candidate data source
2     Apply J48 Decision Tree to learn on the DS and get the decision tree-based
      training model
3     Apply Random Tree to learn on the DS and get the random tree-based training
      model
4     Extract decision trees from both training models
5     return Decision trees
```

Algorithm 6.1: Hybrid with Decision Tree and Random Tree Algorithm

The second mechanism, called rule extraction, acquires decision rules from previously obtained decision trees. To do so, it tracks, in automated fashion, the path from the root node to each leaf node in both trees in order to detach the set of decision rules. The output of this mechanism is a set of decision rules that follow IF-THEN statements. More precisely, a decision rule is a simple IF-THEN statement consisting of an antecedent, is also called the condition, and a consequent, is also called the prediction. For example:

IF <A> (Antecedent) **THEN** <C> (Consequent),

where the antecedent part <A> represents user's surrounding contextual information such as, temporal context, spatial context, social contexts or others relevant contextual information, for example, 'outlook=overcast' and the consequent part <C> represents their corresponding behavioral activities for decision-making, for example, 'play=yes'. It is worth noting that a decision rule can be used to make decisions and predictions.

- **Rule validation.**

  After the completion of the learning-based rule generation step, the rule validation step is accomplished through the following methods:

  - Rule structure verification, which is responsible for checking whether extracted rules are preserving the inherent IF-THEN structure;
  - Rule consistency verification, which is in charge of verifying the consistency of decision rules. Given the fact that decision rules are made up of antecedent

constraints and a consequent constraint, the rule consistency is related to the satisfiability of constraints. To ensure a conflict-free temporary rule repository, a reasoner is used to enumerate all inconsistent rules, where the consequent constraint does not refer to the antecedent constraint.

6.3.1.2 Rule Optimization

The rule optimization is the second and last phase in the rule generation module. The present phase is in charge of identifying the well-performed decision rules from a set of earlier validated rules. In this sense, a GA is extended to support a multi-analysis technique. The main idea of extending a GA with the multi-analysis technique is to optimize the set of decision rules in order to find the well-performed rules by adopting the optimization strategies for GA. To be more precise, there are two main steps in this phase as depicted in Figure 6.3.

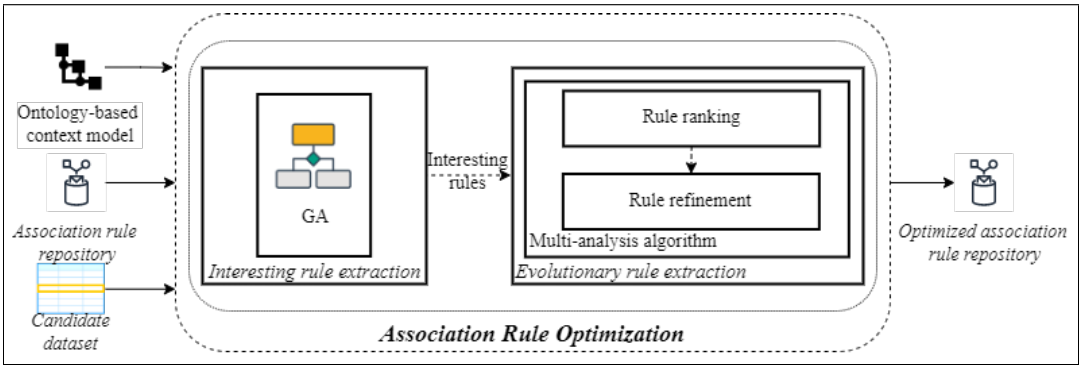

Figure 6.3: Rule optimization phase.

First, the interesting rule extraction step is carried out using GA that is a kind of effective optimizing technique owing to the robust and global search ability [Grefenstette, 1993]. Second, the evolutionary rule extraction step involves a multi-analysis technique to extend the GA and extract the well-performed decision rules.

- **Interesting rule extraction.** This step determines the interesting decision rules, among all discovered decision rules, by applying a GA on the candidate data source. The GA is used to produce a strong level of decision rules since adopted supervised learning algorithms may learn and generate irrelevant rules. The motivation behind this choice is two-fold:

  - First, GA is one of the best methods for rule optimization [Haldulakar and Agrawal, 2011];
  - Second, GA performs a global search technique to find out interesting rules with less complexity compared to other algorithms [Sarath and Ravi, 2013].

  This GA searches for decision rules that are interesting according to a certain rule-interestingness measure. The procedure to select the set of most interesting rules is based on the idea of measuring the interestingness of decision rules that is measured in the fitness function of the GA. For that, the GA creates an initial population as a collection of chromosomes, in which every chromosome represents a decision rule. Then, it evolves the initial population over multiple generations through encoding, selection, crossover and mutation operations to reach the optimal set of interesting decision rules. At the end, it introduces a set of interesting rules, which satisfies a certain measure called fitness function. The flowchart showing the process of GA is given in Figure 6.4.

The flowchart for the proposed GA is as follows:

1. **Encoding.** Candidate data source is encoded to initiate the experimentation of the GA. In our case, a binary encoding schema is used;
2. **Initial population generation.** An initial population of size K chromosomes is randomly generated as a set of solutions to be optimized. These chromosomes are a representation of the rules generated from the candidate data source;
3. **Calculating fitness.** The fitness value of each chromosome in the population is calculated by a fitness function to find the decision rules that their support and confidence are larger than other rules. To this end, a fitness function that has been described in the work of [Qodmanan et al., 2011] is considered as given in Equation 6.1;

$$fitness = \frac{(1 + supp(A \cup C))^2}{1 + supp(A)} \tag{6.1}$$

   In Equation 6.1, supp(A∪C) is the Support of the A -> C and supp(A) is the Support of antecedent part of it;
4. **Selection.** A chromosome with a high fitness value is selected from the population on the basis of a fitness function;
5. **Crossover.** A next generation of population is generated based on the calculated fitness values. The idea behind crossover is to combine the two parent chromosomes to produce two new offspring. The result of crossover is the birth of two new chromosomes. Crossover is carried out according to a defined crossover probability;
6. **Mutation.** Mutation randomly changes chosen bits from 0 to 1 or from 1 to 0. It is applied on the new offspring with a certain mutation probability. The purpose is to maintain diversity among the different generations to increase the global optimization of the GA;
7. **Generation of the next generation.** After a series of selection, crossover and mutation, the GA is stopped when the generated chromosomes meet the optimality or the maximum number of generations. Otherwise, it turns back to stage 3 to continue the rule optimization;

- **Evolutionary rule extraction.**

  In the previous step, the interesting decision rules are selected. However, we cannot get the well-performed rules that could lead to achieve the appropriate decision-making performance due to the fact that fitness function, defined in Equation 6.1, might be in conflict. To deal with this issue, the evolutionary rule step proposes a multi-analysis technique for extending the proposed GA in order to enhance the rule optimization result. For that, rule ranking and rule refinement processes are applied as shown in Figure 6.3. The multi-analysis technique starts with the rule ranking

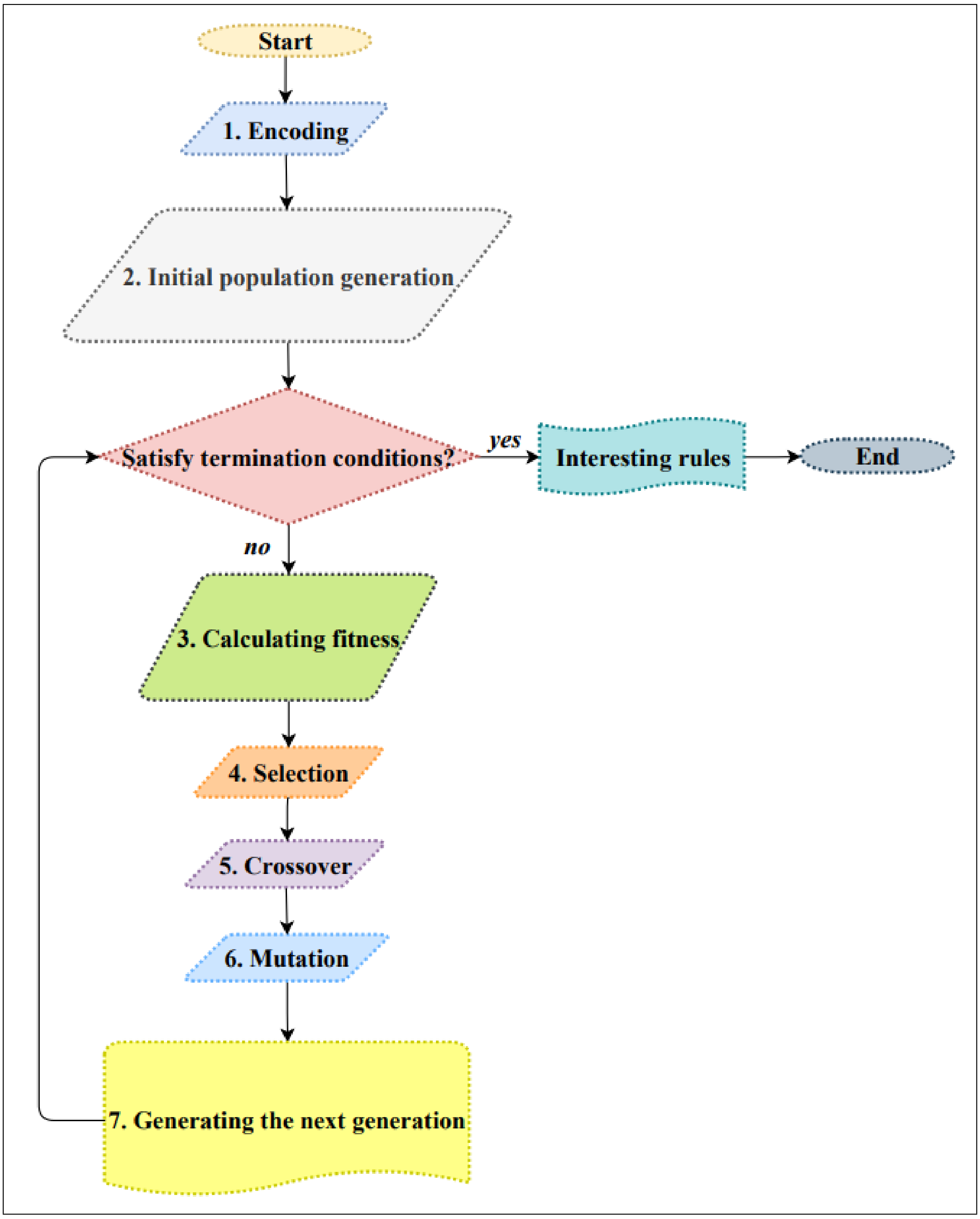

Figure 6.4: Flowchart of a GA.

process that is in charge of automatically analyzing and ranking the interesting decision rules and the generated rules from Machine Learning algorithms regarding their frequency of occurrence and their statistical information. Then, the multi-analysis technique finishes with the proposed rule ranking process. In this process, the rule occurrence frequency is considered as the highest priority to classify rules, followed by the statistical information, such as, the fitness function weight. Then, the rule refinement process is performed to derive the set of well-performed rules. This process begins with finding a user who is related to generated rules to load the user profile of the corresponding user from the ontology-based context model and to infer the well-performed rules that could significantly match with the profile.

6.3.2 Rule Transformation Module

The second module in the proposed DMA approach is the rule transformation module since obtained IF-THEN well-performed decision rules are not enough because these rules should be interpreted and used for decision-making at runtime. Presentation of the obtained decision rules in the evolutionary rule extraction step in a format easily understood, valid and useful is the key in this module. This module aims to transform these IF-THEN rules to Jena rules in order to reason over an ontology-based context model and to infer high-level knowledge using the Jena inference engine. The motivation to opt for Jena is its built-in support for rule-based inference over RDF and OWL [Hussain and Abidi, 2007]. In this sense, the rule transformation keeps track of the well-performed decision rule set and automatically transforms them into Jena rules according to the rule syntax of Jena. Subsequently, rules in Jena format induced from the well-performed decision rules are loaded into the rule knowledge base for an automatic enrichment purpose at runtime. The rule transformation module works during design time and runtime.

At design time, a rule translation is ensured, where a Jena rule definition metamodel, illustrated in Figure 6.5, is proposed to specify the abstract syntax of a Jena rule. A rule set consists of rules. A Jena rule has a name and is composed of concepts. A concept is interpreted as an antecedent and as consequent. Therefore, a Jena rule can contain one or more atoms in the antecedent part and only one atom in the consequent part. Each atom has an attribute, type and value. Then, the proposed metamodel is translated to a Domain Specific Language (DSL) metamodel. This metamodel is defined as textual structure using Xtext. A fragment of the DSL metamodel is depicted in Figure 6.6.

During runtime, the transformation module performs two paramount phases: (i) Instantiation of the DSL metamodel and (ii) Generation of the Jena rules on the basis of the DSL model as previously depicted in Figure 6.1.

- **DSL Rule Instantiation**

  The DSL rule instantiation phase is dedicated to introduce a DSL model, which is a formal rule specification of the defined DSL metamodel in Figure 6.6. The DSL model represents the obtained well-performed decision rules. Here, the rule instantiation is performed automatically at runtime;

- **Jena Rule Generation**

  The Jena rule generation phase is considered for automatically generating the corresponding Jena rules based on the introduced DSL model using Xtend as a trans-

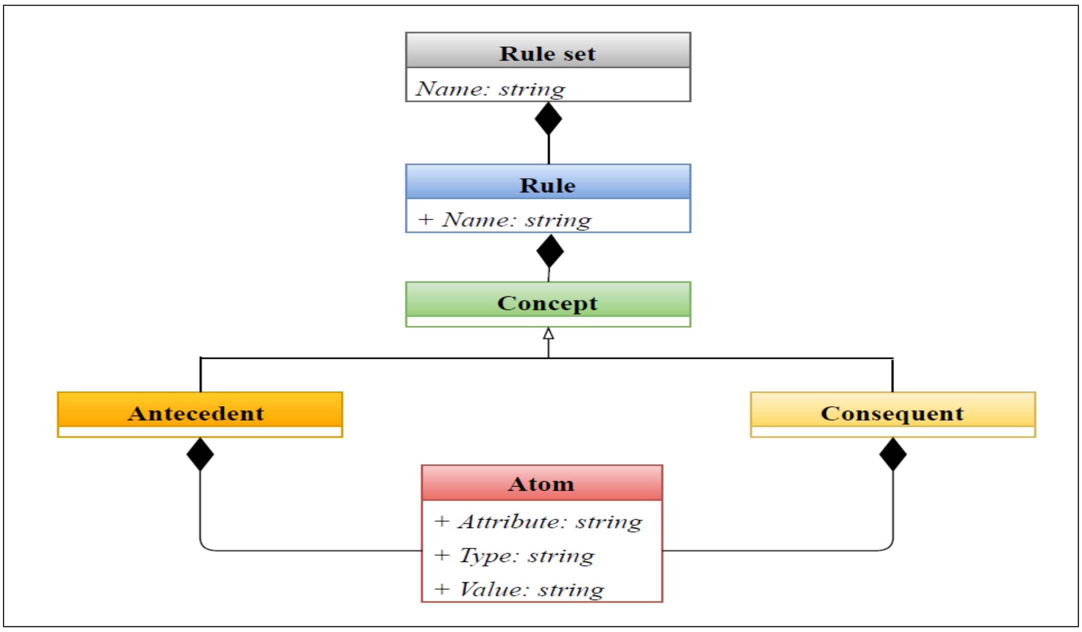

Figure 6.5: Jena rule definition metamodel.

```
grammar org.xtext.example.mydsl.MyDsl with org.eclipse.xtext.common.Terminals

generate myDsl "http://www.xtext.org/example/mydsl/MyDsl"

import "http://www.eclipse.org/emf/2002/Ecore" as ecore

RuleList:
    'rules' name = STRING
    '['
        rules += Rule (',' rules += Rule)*
    ']'
;
Rule:
    'rule' name = STRING
    '{'
        concepts += Concepts (',' concepts += Concepts)*
    '}'
;
Concepts:
    Antecedent | Consequent
```

Figure 6.6: A fragment of the textual description of the DSL metamodel.

formation language. Therefore, a set of Jena rules is generated from the DSL model while the semantic of the DSL metamodel is well-defined.

6.4 DMA Implementation and Case Studies

The previous sections, we have presented the DMA approach to support the automatic enrichment of rule knowledge bases at runtime through learning and generating new decision rules that could reflect changes occurring in dynamic environments and context models. This section begins by describing the prototypical implementation of the present approach and then presenting three case studies.

6.4.1 DMA Implementation

The present DMA approach was designed and implemented to verify its feasibility. The whole architecture implementation will be described, starting from the data source selection module. In the scope of this implementation, we consider an application for assisting engineers and decision makers in evolving a rule knowledge base regarding an ontology-based context model evolution due to arisen changes in the surrounding environment at runtime. In this application, the rule knowledge base, the evolved context model and the data source used in the context evolution process are received as input from the CoE approach. The presented application consists of two distinct parts, called frontend and backend. The backend part, which deals with the automatic rule generation as well as the semi-automatic rule transformation, is implemented as REST web service [Fielding and Taylor, 2002]. The frontend is created with Angular to deal with engineers and decision makers interaction and interfaces with the backend.

In the backend, WEKA, which has a Java open-source library with a great number of Machine Learning algorithms, Xtend and Xtext, which is a framework for developing programming languages and DSLs, were employed for the decision rule generation.

6.4.2 DMA Case Studies

In order to demonstrate the use of our implemented DMA approach in practice and show how it behaved in realistic settings leaves this section presents three case studies provided in the following. It is worth noting that, in these case studies, we consider the same environments' settings and the same users as those in the three scenarios described in section 5.4.2. Thus, the generated rules are carried out using the same HHAR, Ordonez and ARAS data sources.

6.4.2.1 Case Study 1: HHAR Data Source.

In this first case study, we give an example of decision rules generated from the HHAR data source [Stisen et al., 2015] to answer Jean's evolved ontology-based context model at t_1 (see Figure 5.15). HHAR data source contains different attributes, such as the three-accelerometer axis, and one decision attribute is 'Activity'. First, we show, in Table 6.1, a sample of the generated IF–THEN rules obtained through training both Decision Tree and Random Tree algorithms on HHAR data source. Looking more closely at Table 6.1, we can notice that the antecedent part of the rules reflects users' contextual knowledge, and the consequent part represents their associated behavioral activities. Then, we illustrate, in Table 6.2, a sample of the well-performed rules extracted from the generated ones. As illustrated in Table 6.2, the extracted well-performed rules do not differ from Tom to Jean since all activities contained in HHAR data source correlate to their profiles. Finally, after the automatic DSL metamodel instantiation and transformation, we list, in Table 6.3, a sample of the well-performed rules transformed to Jena rules to enrich an existing rule knowledge base and improve the decision-making at runtime.

Rule	IF-THEN rules
R1	IF acc_Y < 4.75 THEN Activity = sit
R2	IF acc_Z <= 3.200101 AND acc_X <= 0.725623 THEN Activity = stand
R3	IF acc_Y >= 4.05 AND acc_Y < 9.63 AND acc_X < 1.13 THEN Activity = stand
R4	IF acc_Y >= 9.63 THEN Activity = walk
R5	IF acc_X > 0.725623 AND acc_Z <= 3.200101 THEN Activity = walk
R6	IF acc_X >= 1.13 AND acc_Y >= 4.05 AND acc_Y < 9.63 THEN Activity = walk
R7	IF acc_X >= 0.29 AND acc_X < 0.32 AND acc_Y < 4.68 THEN Activity = sit
R8	IF acc_X > 0.289698 AND acc_Z <= 10.169373 AND acc_Y <= 4.213394 THEN Activity = sit
R8	IF acc_X > 0.289698 AND acc_Z <= 10.169373 AND acc_Y <= 4.213394 THEN Activity = sit

Table 6.1: A sample of generated rules from the HHAR data source.

Rule	IF-THEN rules	Occurrence frequency	Fitness value
R1.1	IF acc_Y < 4.75 THEN Activity = sit	1	0.745
R1.2	IF acc_Z <= 3.200101 AND acc_X <= 0.725623 THEN Activity = stand	1	0.682
R1.3	IF acc_X > 0.725623 AND acc_Z <= 3.200101 THEN Activity = walk	1	0.682

Table 6.2: A sample of well-performed rules from HHAR data source.

6.4.2.2 Case Study 2: Ordonez Data Source.

In this case study, we describe an example of decision rules generated from the Ordonez data source [Ordónez et al., 2013] to answer Tom and Jean's evolved ontology-based context model at t_2 (see Figure 5.21). As mentioned before, Ordonez data source contains different sensors that are used to track users at their apartments and to predict associated activities like toileting, showering, sleeping, having breakfast, having lunch, having dinner, having snack and so on. First, we list, in Table 6.4, a sample of generated IF–THEN rules obtained through training both Decision Tree and Random Tree algorithms on Ordonez data source. Then, we bring, in Tables 6.5 and 6.6, a sample of the resulting well-performed rules extracted from the generated rules. As we can observe in Tables 6.5 and 6.6, the extracted well-performed rules vary from user to another according to their own profiles. For example, Jean, who is animal-friendly, has a dog in his smart apartment, while Tom, who is not animal-friendly, does not own animals in his own apartment. As a result, the decision rule that stated a grooming activity, is retained for the Jean, who would be asked to groom his or her dog. Finally, after the automatic DSL metamodel instantiation and transformation, we show, in Tables 6.7 and 6.8, a sample of the well-performed rules transformed to Jena rules to enrich an existing rule knowledge base and improve the decision-making at runtime.

Rules	Jena rules
R1.1	[Rule1.1: (?Acc rdf:type uni:hhar:Accelerometer) (?Acc uni:hhar:CheckSup_acc_X ’false’) (?Acc uni:hhar:CheckInf_acc_X ’false’) (?Acc uni:hhar: acc_X ?vx) (?Acc uni:hhar:CheckInf_acc_Y ’true’) (?Acc uni:hhar:ThresholdInf_acc_Y ’4.75’) (?Acc uni:hhar:CheckSup_acc_Y ’false’) (?Acc uni:hhar:acc_Y ?vy) (?Acc uni:hhar:CheckInf_acc_Z ’false’) (?Acc uni:hhar:CheckSup_acc_Z ’false’) (?Acc uni:hhar:acc_Z ?vz) (?Dev rdf:type uni:hhar:Smartphone) (?Acc uni:hhar:sampledBy ?Dev) (?Loc rdf:type uni:conon:Location) (?Dev uni:conon:LocatedIn ?Loc) (?Act rdf:type uni:hhar:Activity) (?Act uni:conon:LocatedIn ?Loc) -> (?Act uni:hhar:label ’sit’)]
R1.2	[Rule1.2: (?Acc rdf:type uni:hhar:Accelerometer) (?Acc uni:hhar:CheckSup_acc_X ’false’) (?Acc uni:hhar:CheckInf_acc_X ’true’) (?Acc uni:hhar:ThresholdInf_acc_X ’0.725623’) (?Acc uni:hhar:acc_X ?vx) (?Acc uni:hhar:CheckInf_acc_Y ’false’) (?Acc uni:hhar:CheckSup_acc_Y ’false’) (?Acc uni:hhar:acc_Y ?vy) (?Acc uni:hhar:CheckInf_acc_Z ’true’) (?Acc uni:hhar:ThresholdInf_acc_Z ’3.200101’) (?Acc uni:hhar:CheckSup_acc_Z ’false’) (?Acc uni:hhar:acc_Z ?vz) (?Dev rdf:type uni:hhar:Smartphone) (?Acc uni:hhar:sampledBy ?Dev) (?Loc rdf:type uni:conon:Location) (?Dev uni:conon:LocatedIn ?Loc) (?Act rdf:type uni:hhar:Activity) (?Act uni:conon:LocatedIn ?Loc) -> (?Act uni:hhar:label ’stand’)]
R1.3	[Rule1.3: (?Acc rdf:type uni:hhar:Accelerometer) (?Acc uni:hhar:CheckSup_acc_X ’true’) (?Acc uni:hhar:ThresholdSup_acc_X ’0.725623’) (?Acc uni:hhar:CheckInf_acc_X ’false’) (?Acc uni:hhar:acc_X ?vx) (?Acc uni:hhar:CheckInf_acc_Y ’false’) (?Acc uni:hhar:CheckSup_acc_Y ’false’) (?Acc uni:hhar:acc_Y ?vy) (?Acc uni:hhar:CheckInf_acc_Z ’true’) (?Acc uni:hhar:ThresholdInf_acc_Z ’3.200101’) (?Acc uni:hhar:CheckSup_acc_Z ’false’) (?Acc uni:hhar:acc_Z ?vz) (?Dev rdf:type uni:hhar:Smartphone) (?Acc uni:hhar:sampledBy ?Dev) (?Loc rdf:type uni:conon:Location) (?Dev uni:conon:LocatedIn ?Loc) (?Act rdf:type uni:hhar:Activity) (?Act uni:conon:LocatedIn ?Loc) -> (?Act uni:hhar:label ’walk’)]

Table 6.3: A sample of well-performed rules in Jena format from HHAR data source.

Rule	IF-THEN rules
R1	IF Place = Living AND Location = Seat AND Type= Pressure THEN Activity = Spare_Time_TV
R2	IF Place = Bathroom AND Location = Shower AND Type= PIR THEN Activity = Grooming
R3	IF Place = Kitchen AND Location = Door AND Type= PIR THEN Activity = Snack
R4	IF Place = Bathroom AND Location = Door AND Type= PIR THEN Activity = Toileting
R5	IF Place = Bedroom AND Location = Bed AND Type= Pressure THEN Activity = Sleeping

Table 6.4: A sample of generated rules from the Ordonez data source.

Rule	IF-THEN rules	Occurrence frequency	Fitness value
R1.1	IF Place = Bedroom AND Location = Bed AND Type= Pressure THEN Activity = Sleeping	5	0.743
R1.2	IF Place = Bathroom AND Location = Door AND Type= PIR THEN Activity = Toileting	3	0.686
R1.3	IF Place = Living AND Location = Seat AND Type= Pressure THEN Activity = Spare_Time_TV	2	0.675
R1.4	IF Place = Kitchen AND Location = Door AND Type= PIR THEN Activity = Snack	1	0. 611
R1.5	IF Place = Bathroom AND Location = Shower AND Type= PIR THEN Activity = Grooming	1	0.521

Table 6.5: A sample of well-performed rules for Jean.

Rule	IF-THEN rules	Occurrence frequency	Fitness value
R2.1	IF Place = Bedroom AND Location = Bed AND Type= Pressure THEN Activity = Sleeping	4	0.743
R2.2	IF Place = Bathroom AND Location = Door AND Type= PIR THEN Activity = Toileting	3	0.686
R2.3	IF Place = Living AND Location = Seat AND Type= Pressure THEN Activity = Spare_Time_TV	2	0.675
R2.4	IF Place = Kitchen AND Location = Door AND Type= PIR THEN Activity = Snack	1	0. 611

Table 6.6: A sample of well-performed rules for Tom.

Rules	Jena rules
R1.1	[Rule1.1: (?placeValue uni:ordonez:Place 'Bedroom') (?locationValue uni:conon:Location 'Bed') (?sensorValue uni:sosa:Sensor 'Pressure') -> (?activity uni:conon:Activity 'Sleeping')]
R1.2	[Rule1.2: (?placeValue uni:ordonez:Place 'Bathroom') (?locationValue uni:conon:Location 'Door') (?sensorValue uni:sosa:Sensor 'PIR') -> (?activity uni:conon:Activity 'Toileting')]
R1.3	[Rule1.3: (?placeValue uni:ordonez:Place 'Living') (?locationValue uni:conon:Location 'Seat') (?sensorValue uni:sosa:Sensor 'Pressure') -> (?activity uni:conon:Activity 'Spare_Time_TV')]
R1.4	[Rule1.4: (?placeValue uni:ordonez:Place 'Kitchen') (?locationValue uni:conon:Location 'Door') (?sensorValue uni:sosa:Sensor 'PIR') -> (?activity uni:conon:Activity 'Snack ')]
R1.5	[Rule1.5: (?placeValue uni:ordonez:Place 'Bathroom') (?locationValue uni:conon:Location 'Shower') (?sensorValue uni:sosa:Sensor 'PIR') -> (?activity uni:conon:Activity 'Grooming')]

Table 6.7: A sample of well-performed rules in Jena format for Jean.

Rules	Jena rules
R2.1	[Rule2.1: (?placeValue uni:ordonez:Place 'Bedroom') (?locationValue uni:conon:Location 'Bed') (?sensorValue uni:sosa:Sensor 'Pressure') -> (?activity uni:conon:Activity 'Sleeping')]
R2.2	[Rule2.2: (?placeValue uni:ordonez:Place 'Bathroom') (?locationValue uni:conon:Location 'Door') (?sensorValue uni:sosa:Sensor 'PIR') -> (?activity uni:conon:Activity 'Toileting')]
R2.3	[Rule2.3: (?placeValue uni:ordonez:Place 'Living') (?locationValue uni:conon:Location 'Seat') (?sensorValue uni:sosa:Sensor 'Pressure') -> (?activity uni:conon:Activity 'Spare_Time_TV')]
R2.4	[Rule2.4: (?placeValue uni:ordonez:Place 'Kitchen') (?locationValue uni:conon:Location 'Door') (?sensorValue uni:sosa:Sensor 'PIR') -> (?activity uni:conon:Activity 'Snack ')]

Table 6.8: A sample of well-performed rules in Jena format for Tom.

6.4.2.3 Case Study 3: ARAS Data Source.

In this case study, we take an example of decision rules generated from the ARAS data source [Alemdar et al., 2013] to answer Tom and Jean's evolved ontology-based context model at t_3 (see Figure 5.34). The ARAS data source, where sensors are used to track users at their homes and predict associated activities, contains several features, such as photocell, force, distance, contact, sonar distance, temperature and IR sensors, and one decision attribute is 'Activity'. The candidate data source features are binary sensor readings as 1 when the sensor is activated and 0 when the sensor is deactivated. First, we show, in Table 6.9, a sample of generated IF–THEN rules obtained through training both Decision Tree and Random Tree algorithms on ARAS data source. Then, we illustrate, in Tables 6.10 and 6.11, a sample of well-performed rules extracted from the generated rules. As illustrated in Tables 6.10 and 6.11, the extracted well-performed rules vary from Tom to Jean according to their own profiles. For example, Tom prefers in his spare time to read a book as an entertainment form, while his young son Jean prefers music as an entertainment form. As a result, decision rules that stated a spare time activity, where the distance sensor is deactivated (e.g., Distance=0) and the force sensor is activated (e.g., ForceSensor=1), and a user could have fun listening to music (Activity=ListeningToMusic), are overlooked for the Tom, who shows more interest in reading books in his spare time. Finally, after the automatic DSL metamodel instantiation and transformation, we list, in Tables 6.12 and 6.13, a sample of the well-performed rules transformed to Jena rules to enrich an existing rule knowledge base and improve the decision-making at runtime.

Rule	IF-THEN rules
R1	IF Distance=1 AND ForceSensor=0 AND Couch=1 THEN Activity=HavingDinner
R2	IF Distance=0 AND Temperature=0 AND ForceSensor=1 AND Bed=1 THEN Activity=Sleeping
R3	IF Distance=1 AND ForceSensor=1 AND Couch=1 THEN Activity=Studying
R4	IF Distance=0 AND ForceSensor=1 AND Couch=1 THEN Activity= ListeningToMusic
R5	IF Distance=1 AND ForceSensor=1 AND Temperature=0 AND Chair=1 THEN Activity= UsingInternet
R6	IF Distance=0 AND ForceSensor=1 AND Couch=1 THEN activity=ReadingBook
R7	IF Distance=1 AND ForceSensor=0 AND Temperature=1 AND KitchenDrawer=1 THEN Activity= PreparingDinner

Table 6.9: A sample of generated rules from the ARAS data source.

Rule	IF-THEN rules	Occurrence frequency	Fitness value
R1.1	IF Distance=1 AND ForceSensor=0 AND Couch=1 THEN Activity=HavingDinner	7	0.743
R1.2	IF Distance=0 AND Temperature=0 AND ForceSensor=1 AND Bed=1 THEN Activity=Sleeping	6	0.686
R1.3	IF Distance=1 AND ForceSensor=1 AND Temperature=0 AND Chair=1 THEN Activity= UsingInternet	3	0.660
R1.4	IF Distance=0 AND ForceSensor=1 AND Couch=1 THEN activity=ReadingBook	2	0.660

Table 6.10: A sample of well-performed rules for Tom.

Rule	IF-THEN rules	Occurrence frequency	Fitness value
R2.1	IF Distance=1 AND ForceSensor=0 AND Couch=1 THEN Activity=HavingDinner	7	0.743
R2.2	IF Distance=0 AND Temperature=0 AND ForceSensor=1 AND Bed=1 THEN Activity=Sleeping	6	0.686
R2.3	IF Distance=1 AND ForceSensor=1 AND Couch=1 THEN Activity=Studying	3	0.611
R2.4	IF Distance=0 AND ForceSensor=1 AND Couch=1 THEN Activity= ListeningToMusic	1	0.521

Table 6.11: A sample of well-performed rules for Jean.

Rules	Jena rules
R1.1	[ruleR1.1: (?DistanceValue uni:aras:Distance '1') (?ForceSensorValue uni:aras:ForceSensor '0') (?CouchValue uni:aras:Couch '1') -> (?ActivityValue uni:conon:Activity 'HavingDinner')]
R1.2	[ruleR1.2: (?DistanceValue uni:aras:Distance '0') (?ForceSensorValue uni:aras:ForceSensor '1') (?TemperatureValue uni:aras:Temperature '0') (?BedValue uni:aras:Bed '1') -> (?ActivityValue uni:conon:Activity 'Sleeping')]
R1.3	[ruleR1.3: (?DistanceValue uni:aras:Distance '1') (?ForceSensorValue uni:aras:ForceSensor '1') (?TemperatureValue uni:aras:Temperature '0') (?ChairValue uni:aras:Chair '1') -> (?ActivityValue uni:conon:Activity 'UsingInternet')]
R1.4	[ruleR1.4: (?DistanceValue uni:aras:Distance '0') (?ForceSensorValue uni:aras:ForceSensor '1') (?CouchValue uni:aras:Couch '1') -> (?ActivityValue uni:conon:Activity 'ReadingBook')]

Table 6.12: A sample of well-performed rules in Jena format for Tom.

Rules	Jena rules
R2.1	[ruleR2.1: (?DistanceValue uni:aras:Distance '1') (?ForceSensorValue uni:aras:ForceSensor '0') (?CouchValue uni:aras:Couch '1') -> (?ActivityValue uni:conon:Activity 'HavingDinner')]
R2.2	[ruleR2.2: (?DistanceValue uni:aras:Distance '0') (?ForceSensorValue uni:aras:ForceSensor '1') (?TemperatureValue uni:aras:Temperature '0') (?BedValue uni:aras:Bed '1') -> (?ActivityValue uni:conon:Activity 'Sleeping')]
R2.3	[rule R2.3: (?DistanceValue uni:aras:Distance '1') (?ForceSensorValue uni:aras:ForceSensor '1') (?CouchValue uni:aras:Couch '1') -> (?ActivityValue uni:conon:Activity 'Studying')]
R2.4	[ruleR2.4: (?DistanceValue uni:aras:Distance '0') (?ForceSensorValue uni:aras:ForceSensor '1') (?CouchValue uni:aras:Couch '1') -> (?activityValue uni:conon:Activity 'ListeningToMusic')]

Table 6.13: A sample of well-performed rules in Jena format for Jean.

6.5 Concluding Remarks

This chapter has provided the DMA approach to present how decision rules can be generated from candidate data sources to reflect changes in dynamic environments without having to consider human intervention at runtime. In this chapter, we have described the second approach into this path: the DMA approach of the IConAS approach. Additionally, an implementation of the DMA approach has been presented. Moreover, three case studies have been carried out and discussed. Therefore, this chapter has presented a detailed illustration of the DMA approach as follows:

- Firstly, in section 6.2, we have provided an overview of the decision-making adaptation approach;
- Secondly, in section 6.3, we have introduced the general architecture of our approach. Then we have described the key modules defined in this architecture;

- Finally, in section 6.4, we have first proposed an implementation of our approach. Then, we have described three case studies.

Further evaluations of DMA approach will be conducted in chapter 7. They will be discussed in section 7.3.

CHAPTER 7

Evaluation

7.1 Introduction

This chapter deals with the evaluation studies performed to validate the research work presented in this thesis. More precisely, the main goal of these studies is to validate and assess the proposed CoE and DMA approaches. In addition, a secondary goal is to assess the proposed IntElyCare framework, describing the IConAS approach in the healthcare domain. These studies address the research questions RQ.1.3. and RQ.2.2. as well as the case studies described in chapter 1, chapter 5 and chapter 7, respectively.

For fulfilling these studies, this chapter is organized under different sections:

- Firstly, section 7.2 is conducted to assess CoE using feature-based, criteria-based, expert-based and competency question-based evaluation approaches.
- Next, section 7.3 is used to evaluate the effectiveness of DMA in terms of number of rules, performance and computational time.
- Then, section 7.4 is devoted to evaluate the IConAS approach through an elderly healthcare case study together with activity recognition and user satisfaction evaluation approaches.
- Finally, section 7.5 concludes this chapter by summarizing the obtained evaluation findings.

7.2 CoE Evaluation

According to Suárez-Figueroa [Suárez-Figueroa, 2010], the process of ontology evaluation is composed of two major activities, namely, ontology verification and ontology validation. These activities seek to identify, respectively, whether the evaluated ontology is a correct ontology and whether it is produced in the right way. First, the ontology verification activity relates to the correctness of the ontology and especially investigates its structure, functionality and representation with the help of different metrics and quality criteria. Second, the ontology validation activity relates to the question of how much an ontology is well-founded and corresponds accurately to the environment it represents. In the ontology validation activity, the meaning of the definitions is compared against the environment that the ontology has been developed or evolved to represent.

In this thesis, we followed two major activities to evaluate the evolved ontology-based context models. For that, we defined the verification as the process of checking that evolved ontology-based context models meet specifications of ontology optimization and quality, while the validation as the process of checking whether evolved ontology-based context models accurately represent and capture the target environment. The proposed evaluation is four-fold, where feature-based, criteria-based, expert-based and competency question-based evaluation approaches are applied. Therefore, many methods and tools, such as OntoMetrics, Ontology Pitfall Scanner (OOPS!) [Poveda-Villalón et al., 2014] and competency questions are included. To perform the evaluation process of our CoE approach, we assessed different evolved ontology-based models' results gained from the presented case studies in chapter 5 and from other case studies that are not covered in this thesis. Table 7.1 shows the number of concepts, the number relations, and the information about the candidate data sources used in the different evolutions.

	#Concepts	#Relations	Candidate data source			
			Name	**Setting**	**Size**	**Observation**
Evolved ontology-based model 1	68	80	ARAS[1]	Smart home	47	5,184,000
Evolved ontology-based model 2	37	54	Ordonez[2]	Smart apartment	24	20,358
Evolved ontology-based model 3	26	38	HHAR[3]	Ordinar apartment	16	43,930,250
Evolved ontology-based model 4	22	31	RCD[4]	Smart room	14	250,000
Evolved ontology-based model 5	139	167	ExtraSensory2[5]	Outdoor or indoor	127	300,000

[a][Alemdar et al., 2013]
[b][Ordónez et al., 2013]
[c][Stisen et al., 2015]
[d][Morgner et al., 2017]
[e][Vaizman et al., 2017]

Table 7.1: Insights in the evolved ontology-based models.

7.2.1 Feature-based Evaluation

For the ontology verification activity, we used a feature-based evaluation that is oriented toward evaluating the structural and schema aspects of evolved ontologies to draw conclusions about their structure and schema quality. This evaluation has been adopted by different techniques, such as OntoClean [Guarino and Welty, 2002], OntoQA [Tartir et al., 2005], and OntoMetrics [Lantow, 2016]. For our evolved ontology-based context models, OntoMetrics was chosen as a feature-based evaluation framework. It is a Web-based tool that validates and computes the ontology structure and schema quality metrics, such as base, schema, class, knowledge base, and graph metrics. In this evolution, we aimed to reuse two main metrics from OntoMetrics, called schema metrics and graph metrics, to determine the structure and schema quality with respect to the concepts, relations, and inheritance levels of the evolved ontology-based context models. An overview of these metrics is discussed as follows:

- ***Schema metrics*** assess the design of ontology by indicating the richness, width, depth and inheritance of an ontology schema design. For this, they calculate and compare statistics about the concepts, inheritance levels, relation types, properties, and other elements. Thus, schema metrics stand for searching for schema-related errors such as recursive definitions, unconnected concepts, missing domains or ranges and missing inverse relations. The most essential and significant metrics of the schema category are the following: the Inheritance Richness (IR), Relationship Richness (RR) and Inverse Relations Ratio (IRR).
  - ***IR*** is a way to measure the overall levels of the distribution of concepts of the ontology's inheritance tree. It is a good indication of how well concepts are grouped into different categories and subcategories in the ontology. It is known as the average number of sub-concepts per concept to describe how the concepts are distributed across the different levels of the ontology and thus distinguish shallow ontologies, where concepts have a large number of direct sub concepts, from deep ontologies, where concepts have a small number of direct sub concepts. A relatively low IR result would correspond to a deep (or vertical) ontology that covers its targeted environment in a very detailed manner, while a high IR result, by contrast, would reflect a shallow (or horizontal) ontology that tends to represent a wide range of general concepts with fewer level of detail.
  - ***RR*** examines the existing relations within an ontology to reflect the diversity of relationships. An ontology that contains only inheritance relationships usually conveys less information than an ontology that contains a diverse set of relationships. It is represented as the fraction of the number of non-taxonomic relations, specifically the object properties, and the total number of relations that can include sub-concepts and non-taxonomic relations in the ontology. The RR result is a number between 0 and 1, where a high value closer to 1 indicates that the ontology is rich and contains a variety of non-taxonomic relations, while a small RR value closer to 0 indicates that the ontology mostly consists of subsumption relations.
  - ***IRR*** illustrates the ratio between the inverse non-taxonomic relations and all non-taxonomic relations. Lower values for this metric indicate a deficiency in the definition of inverse non-taxonomic relations in the ontology.
- ***Graph metrics*** are also known as structural metrics, where the structure (or taxonomy) of ontologies is analyzed. These metrics calculate the cardinality and depth of the ontology structure in terms of the absolute and average depth, breadth, and so on. The depth metric that consists of an absolute and average is associated with the cardinality of the paths. It is a property of graphs which is related to the cardinality of paths existing in the graph. The arcs which are considered are only is-a arcs but

this only applies to directed graphs. The breadth metric, which is represented by the absolute and average, expresses the cardinality of the levels. The value of these different parameters in the graph metrics depicts the effectiveness of an ontology structure.

With regard to the obtained evolved models, the mean results of the schema metrics are given in Figure 7.1. From Figure 7.1, it can be seen that we achieved a mean value for the inheritance richness IR equal to 0.903. As IR has been proposed to distinguish a deep ontology from a shallow ontology, the obtained high IR result proved that the evolved models are deep ontologies. These models might reflect vertical nature and offer several levels of inheritance, where each concept had at least two sub-concepts. Aside from this, the mean result of the RR was about 0.081. We can note that this result is close to zero, indicating that most relations defined in the evolved models are subsumption relations. It is obvious that the evolved models brought minimal non-taxonomic relations, which could be determined from the mean result of the RR. Furthermore, the obtained mean result of the IRR was equal to 0, which means there is a deficiency of inverse relations in the evolved models.

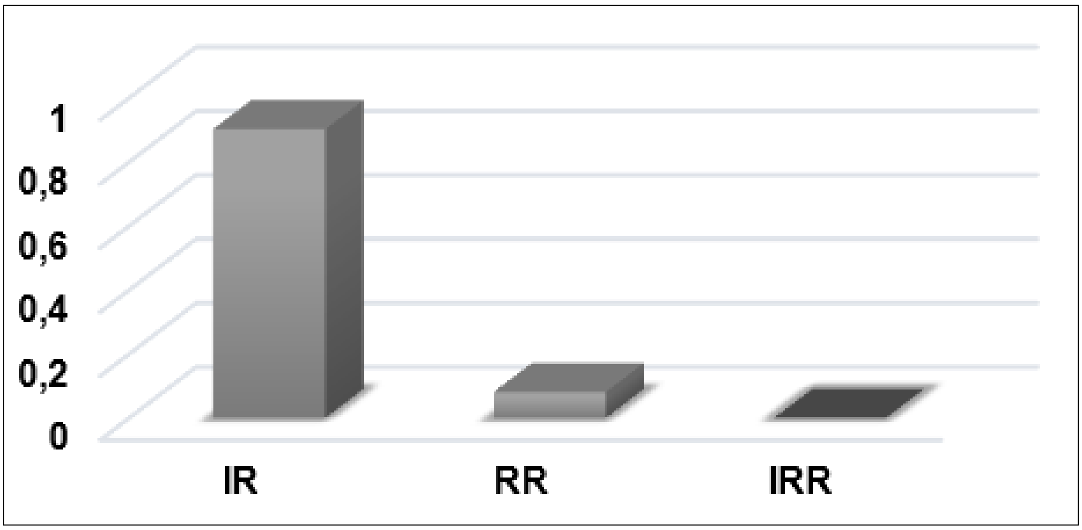

Figure 7.1: Mean results of schema metrics for the evolved ontology-based models using the OntoMetrics tool.

The mean results of the graph metrics are scattered in Figure 7.2. As this figure shows, the evolved models had a mean absolute depth of 135 and a mean average depth of 2.177. Thus, the depth metric results obtained can confirm the verticality of the evolved models as demonstrated by previously reported mean IR result. In turn, the evolved models presented a mean absolute breadth of 62 and a mean average breadth of 7.75. Thus, the breadth metric results reinforce the vertical hierarchical design of the evolved models.

7.2.2 Criteria-based Evaluation

Similarly, for the ontology verification activity, we adopt a criteria-based evaluation that supports the evaluation of an ontology quality. Regarding ontology quality evaluation, Poveda-Villalón, Suárez-Figueroa and Gómez-Pérez [Poveda et al., 2009, Poveda Villalon et al., 2010] provide a set of common worst practices in ontology development and engineering. For

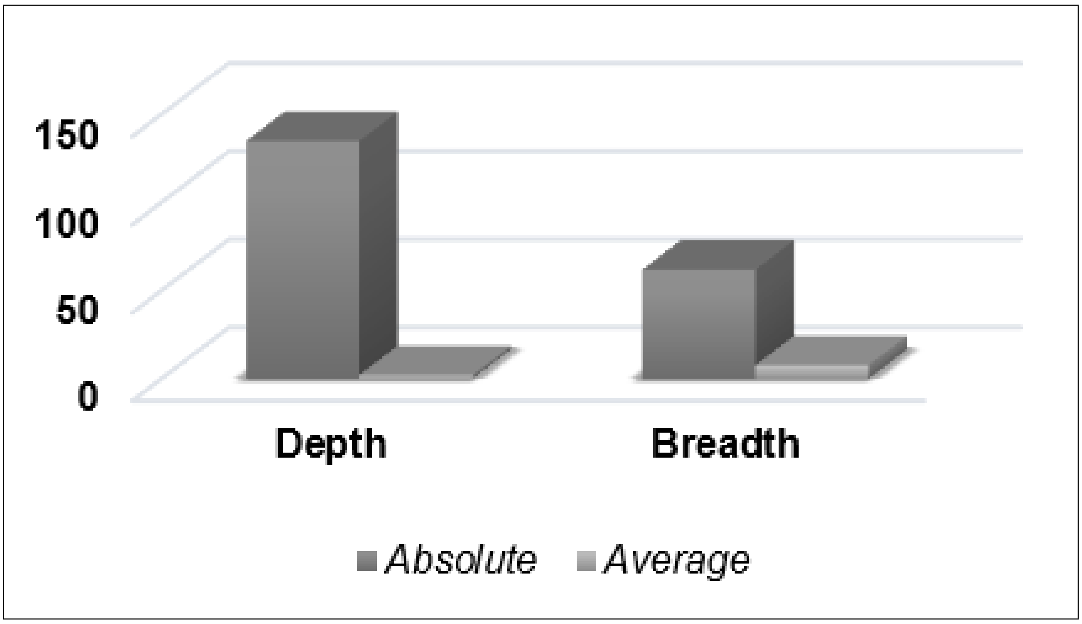

Figure 7.2: Mean results of graph metrics for the evolved ontology-based models using the OntoMetrics tool.

this, they proposed a catalog of pitfalls to evaluate the quality of an ontology against two different classifications. The first classification considers structural, functional and usability-profiling dimensions. Originally, these dimensions were defined by [Gangemi et al., 2005]. The second classification, provided by [Poveda Villalón, 2016], follows three Cs criteria of ontology evaluation proposed by [Gómez-Pérez, 2004], namely: Consistency, Completeness and Conciseness. These criteria, presented in Table 7.2 with their descriptions and their related pitfalls, are an extension to the dimensions of [Gangemi et al., 2005]. It is noteworthy that in the case of the second classification, there are pitfalls that cannot match these three criteria because they do not approach the ontology quality. We notice that the pitfalls affect the ontology in different importance levels, and each level demand a different action [Poveda Villalón, 2016]:

- ***Critical pitfalls*** that give rise to severe problems affecting the consistency or reasoning of the ontology and are mandatory to be corrected.
- ***Important pitfalls*** that refer to not critical problems to the consistency of the ontology but are considered important to correct.
- ***Minor pitfalls*** that do not cause any severe problems, but their correction can make ontology richer and more understandable.

This thesis focuses on dealing with the evaluation of the evolved models' quality against the three Cs using OOPS! tool, which is a great contribution of [Poveda Villalón, 2016]. OOPS! is a web-based evaluation tool for evaluating ontologies against a set of common pitfalls. It generates a comprehensive list of the pitfalls as a result.

In the following, a Web-based evaluation via OOPS! was performed and we provide a brief description about appearing pitfalls. Table 7.3 points out the different pitfalls that were encountered in the evolved models, their frequency along with their specific descriptions. As reported in Table 7.3, the detected pitfalls did not affect the consistency or conciseness of the evolved models. On the contrary, OOPS! showed normal consistency and conciseness in the evolved models. Thus, the evolved models met both of these standards because they did not contain irrelevant or redundant output results and did not include any inconsistencies. By contrast, for completeness where the requirement of the inverse relationships is required [Chansanam et al., 2021], a minor pitfall (P13) was returned regarding the inverse relationships not being explicitly declared. This pitfall revealed that the evolved models omitted the declaration of inverse relationships. Stated differently, it occurs when any relationship does not have an inverse relationship (owl:inverseOf) defined within the ontology model. This is because all relationships, apart from the subsumption ones, could have an inverse relationship in the standard [Poveda Villalón, 2016]. Thus, we can observe that the completeness pitfall result correlated with the above-discussed mean result of the IRR to show that the evolved models are not complete. To fix the detected pitfall, the OOPS! guidelines suggested explicitly declaring inverse relationships in the evolved models.

Criteria	Description	Related Pitfalls
Consistency	To ensure that the evolved ontology-based models do not contain any inconsistencies (e.g., contradictory or conflicting output results).	P05: Define incorrect inverse relationship. P06: Involve cycles in hierarchy. P07: Merging dissimilar concepts in the same concept. P19: Swapping intersection and union. P24: Using recursive definition.
Completeness	To ensure that all output results that are supposed to be in the evolved ontology-based models are explicitly presented.	P04: Creating unconnected ontology elements. P10: Missing disjointness P11: Missing domain or range in properties. P12: Missing equivalent properties. P13: Inverse relationships not explicitly declared.
Conciseness	To ensure that the evolved ontology-based models do not include redundancies (e.g., irrelevant or redundant output results).	P02: Creating class synonyms. P03: Creating "is" relationship place of "rdfs:subClassOf", "rdf:type", or "owl:sameAs". P21: Using a miscellaneous concept.

Table 7.2: Three Cs criteria for evolved ontology-based models' quality evaluation.

7.2.3 Expert-based Evaluation

For the ontology validation activity, we conduct an expert-based evaluation where the ontologies' content is judged on the basis of expert opinion. While this is a subjective evaluation approach, it is frequently considered a good validation process because it relies on the deep knowledge of external experts who can explore the content of an ontology. In our case to provide a ground for evaluating evolution results, expert-based evaluation is more about assessing the content of ontology-based context models' evolution. Stated differently, we built expert-based evaluation to compare the initial and evolved ontology-based context models in terms of coverage of the current surrounding environment. We took

Criteria	Detected Pitfall	Affect to	OOPS! Importance Level	Satisfaction
Consistency	-	-	Normal	Yes, no contradictory or conflicting output results can be inferred by reasoners since OOPS! shows no errors for all evolved models.
Completeness	P13: Inverse relationship not explicitly stated	Non-taxonomic relations	Minor	No, the evolved models are not completed well since inverse relationships were not explicitly defined as determined by OOPS!
Conciseness	-	-	Normal	Yes, no unnecessary or redundant output results were contained in the evolved models according to OOPS!

Table 7.3: Pitfalls in evolved ontology-based models detected by OOPS!

into account the coverage of concepts to capture the sufficiency of the ontology concepts for representing the runtime changes occurring in the current surrounding environment. To this end, a total of 20 expert participants were invited and asked to judge their own level of expertise in (i) ontology development (ii) ontology engineering in general, and (iii) domain. We used a Likert scale with values from 1 (beginner) to 10 (expert).

In this evaluation approach, the initial and evolved ontology-based context models, together with a description of the surrounding environment changes, were given to the invited expert participants, who then applied their knowledge to assess the coverage of the given ontology-based models by checking all their concepts and identifying uncovered concepts to conclude their coverage in response to the emerging changes in the dynamic environment. This evaluation consisted of calculating three well-known metrics:

- Precision was used to indicate how accurately the concepts identified in an ontology-based model represented the current surrounding environment, and it was the number of correct concepts in the ontology-based model relative to the total number of concepts in the ontology-based model, as shown in Equation 7.1.

$$Precision = \frac{Nb\ of\ correct\ concepts\ in\ the\ model}{Total\ Nb\ of\ concepts\ in\ the\ model} \tag{7.1}$$

- Recall was used to measure the environment coverage of the ontology-based model, and it was the number of correct concepts relative to the total number of possible concepts, as shown in Equation 7.2.

$$Recall = \frac{Nb\ of\ correct\ concepts\ in\ the\ model}{Total\ Nb\ of\ possible\ concepts} \tag{7.2}$$

- F1-score was used to measure the accuracy of the ontology-based model, and it was the harmonic mean that combined both the precision and recall values as shown in Equation 7.3.

$$F1 - score = 2\,\frac{Precision\ Recall}{Precision + Recall} \tag{7.3}$$

In the first round of the expert-based evaluation, the invited experts explored the changes occurring in the surrounding environment and then navigated across the initial and evolved ontology-based models. In the second round, they evaluated both models' quality in terms of environment coverage. Consequently, the precision, recall and F1-score mean of the initial and evolved ontology-based models were computed as shown in Table 7.4.

	Precision (%)	Recall (%)	F1-score (%)
Initial ontology-based models	69	48	57
Evolved ontology-based models	80	65	72

Table 7.4: Initial and evolved ontology-based context models' coverage results.

Considering the results presented in Table 7.4, the evolved models achieve average results 80% as precision and 65% as Recall. In addition, they reach a quite good result for the F1-score of 72%, knowing that the F1-score's best value is at 1 and its worst value is at 0. As a whole, the evolved model's coverage results are promising and show considerable precision, Recall and F1-score values. In addition, it is clear that the coverage of evolved context models is better than initial models. According to these results, we observe that the evolved models could fit well with the surrounding environment changes, since they showed a higher coverage level than the initial model.

7.2.4 Competency Question-based Evaluation

One of the ways to ensure ontology validation activity is to sketch a list of questions that the ontology models should be able to answer. These are called Competency Questions (CQs) and expressed in natural language. These CQs will serve as the litmus test in the ontology evaluation. In a competency question-based evaluation, ontology is judged on the basis of CQs that will be used to test the internal validity of the ontology models. If the ontology model is capable of correctly answering CQs with its necessary and sufficient axioms, then the ontology model is validated. In our case, with a set of CQs at hand, it is possible to determine whether an ontology model was evolved correctly according to captured changes in surrounding environments at runtime. As a key objective of ontology-based context evolution is to provide more extensive knowledge, each CQ is expected to have more complete answers using the evolved ontology-based context model compared to the initial ontology-based context model. In fact, we place a set of CQs, which should be answerable once an ontology model has been evolved, based on the captured changes explored in the Scenarios 5.4.2.2 and 5.4.2.2 set out in detail as part of case studies in chapter 5. These CQs are broken down into distinct groups that cover each ontology module presented in chapter 4. Due to space constraints, however, the thesis does not outline every group of the CQs here, but rather discusses and focuses on the two key groups that are represented in mind map diagrams as depicted in Figures 7.3 and 7.4.

In order to enable the automatic evaluation, CQs need to be formalized in a query language with the aim of assessing an ontology model evolution. Therefore, each CQ posed has been converted to a SPARQL query to run against the initial ontology-based context model and evolved ontology-based context models. SPARQL query language is particularly pertinent for CQs to indicate whether these questions can be successfully implemented. To check the ontology-based context models against the CQs in this thesis, we make use of

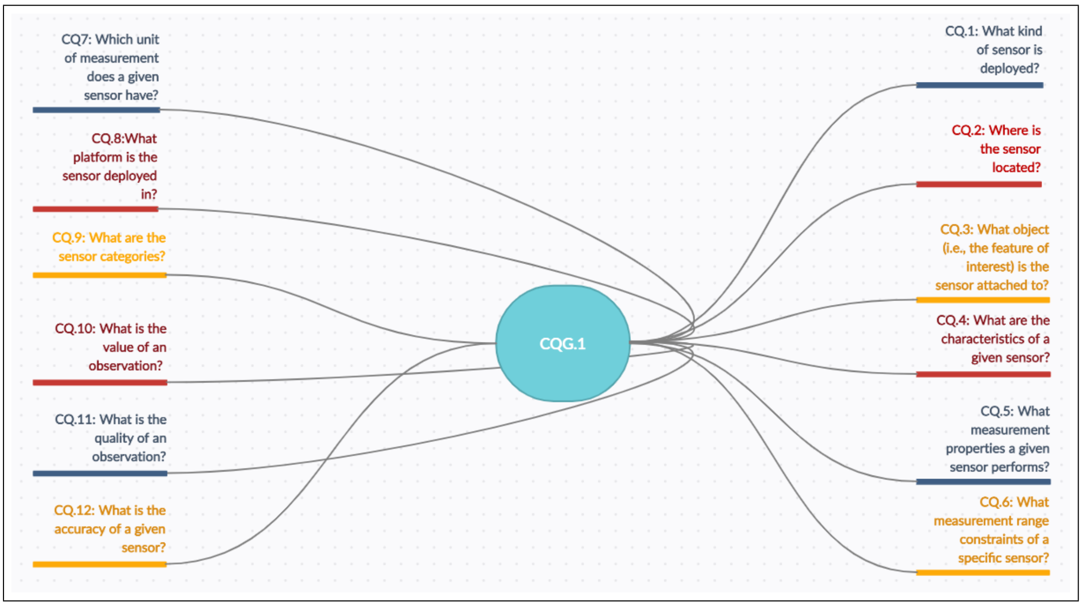

Figure 7.3: CQs regarding sensors.

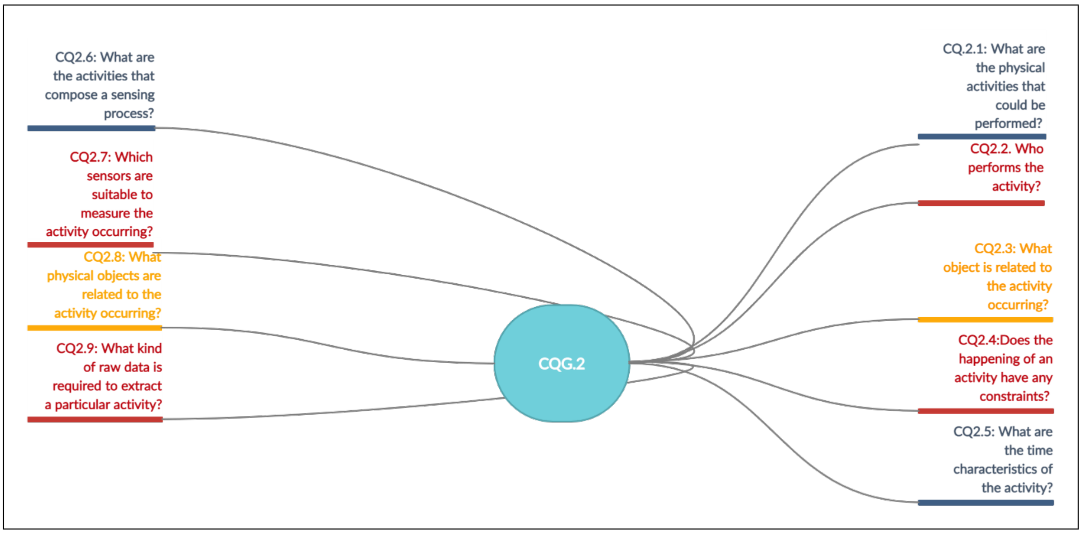

Figure 7.4: CQs regarding human activities.

the different initial ontology models and evolved ontology models that are obtained from Scenarios 5.4.2.2 and 5.4.2.2 and illustrated in Figures 5.17, 5.21 and 5.34, respectively, as:

- O_initial : The initial ontology model, which was achieved by human intervention.
- O_evolved: The evolved ontology model, which was achieved by applying the proposed CoE approach.

The results of running four of these CQs and answers retrieved using SPARQL query are presented in Figures 7.5, 7.8, 7.6, 7.9, 7.7, 7.10, 7.11, 7.12, 7.13, 7.14, 7.15 and 7.16. In addition, Table 7.1 shows the correspondences between the four CQs and their associated answers in more details.

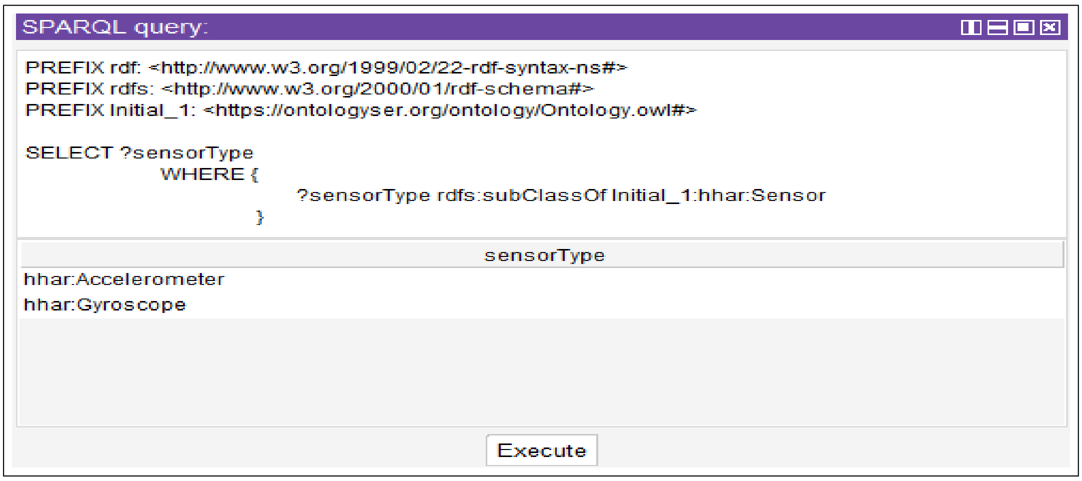

Figure 7.5: SPARQL query results for CQ1.1 at t_1.

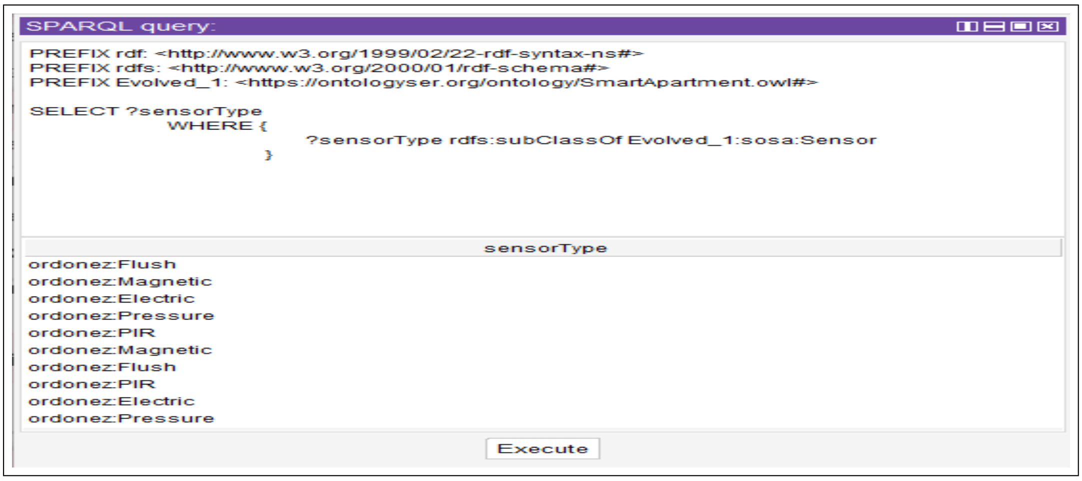

Figure 7.6: SPARQL query results for CQ1.1 at t_2.

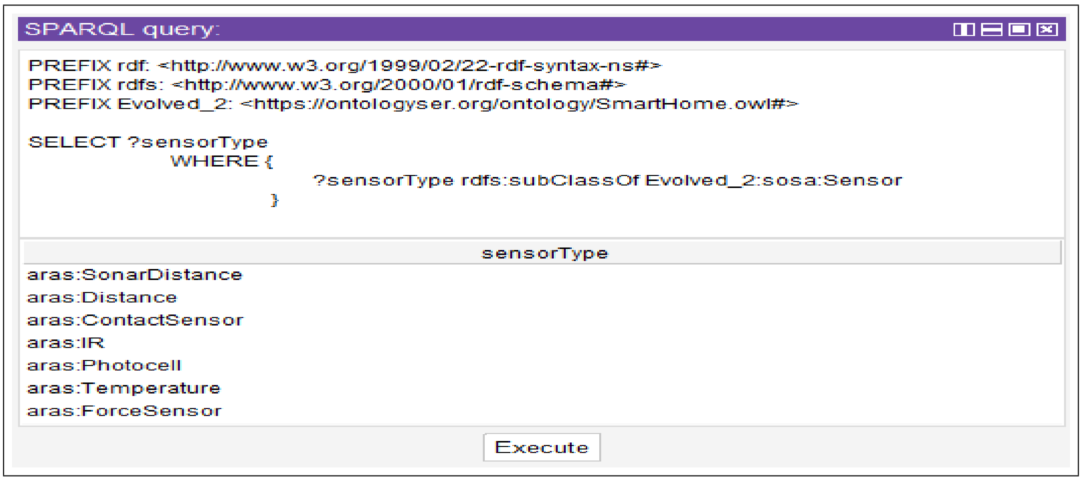

Figure 7.7: SPARQL query results for CQ1.1 at t_3.

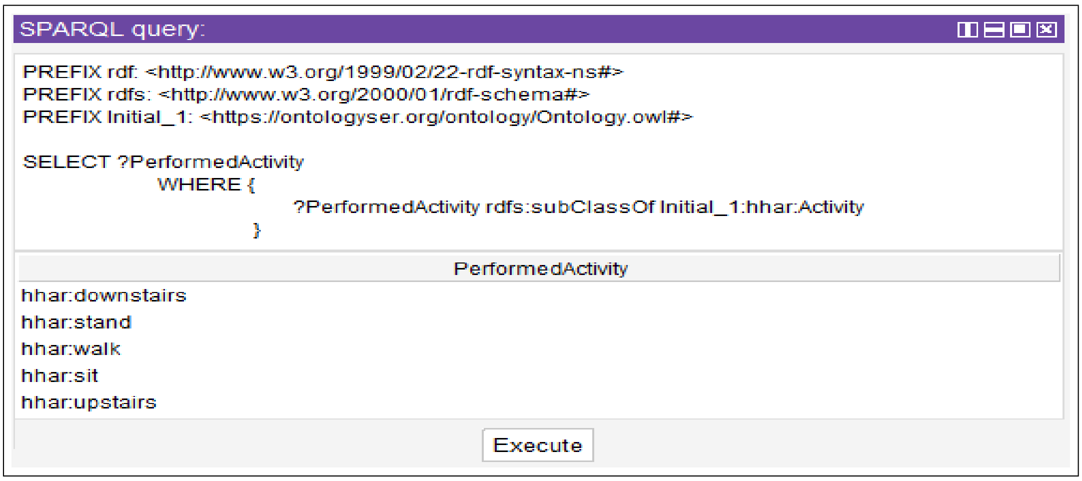

Figure 7.8: SPARQL query results for CQ2.1 at t_1.

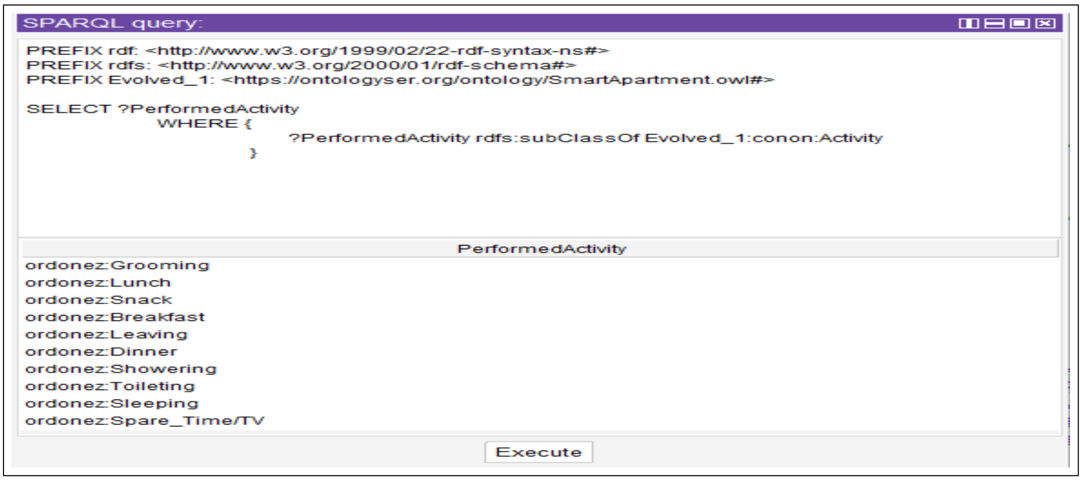

Figure 7.9: SPARQL query results for CQ2.1 at t_2.

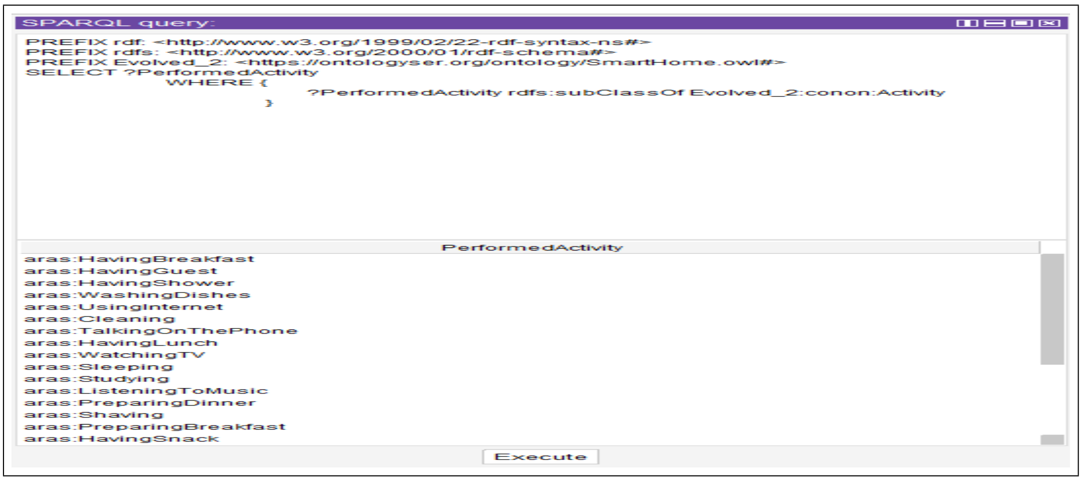

Figure 7.10: SPARQL query results for CQ2.1 at t_3.

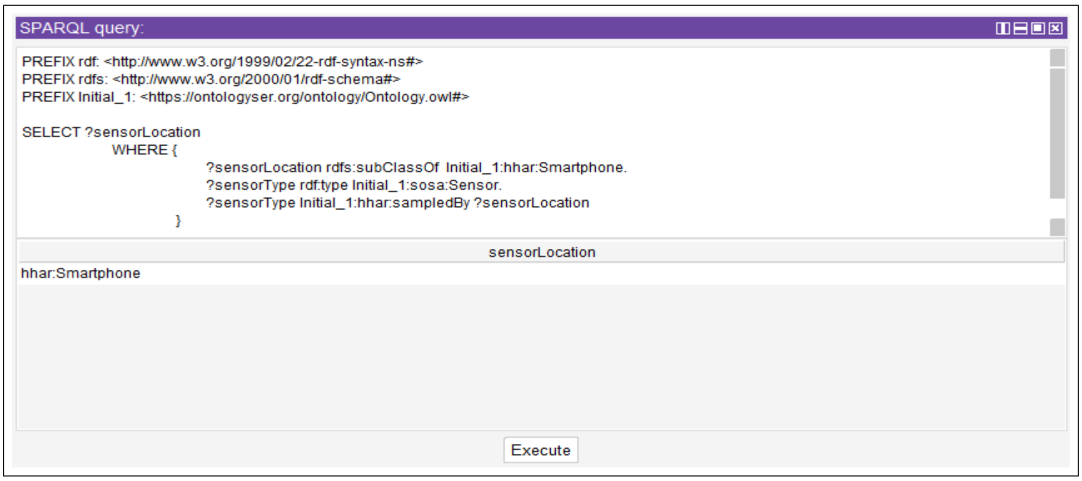

Figure 7.11: SPARQL query results for CQ1.2 at t_1.

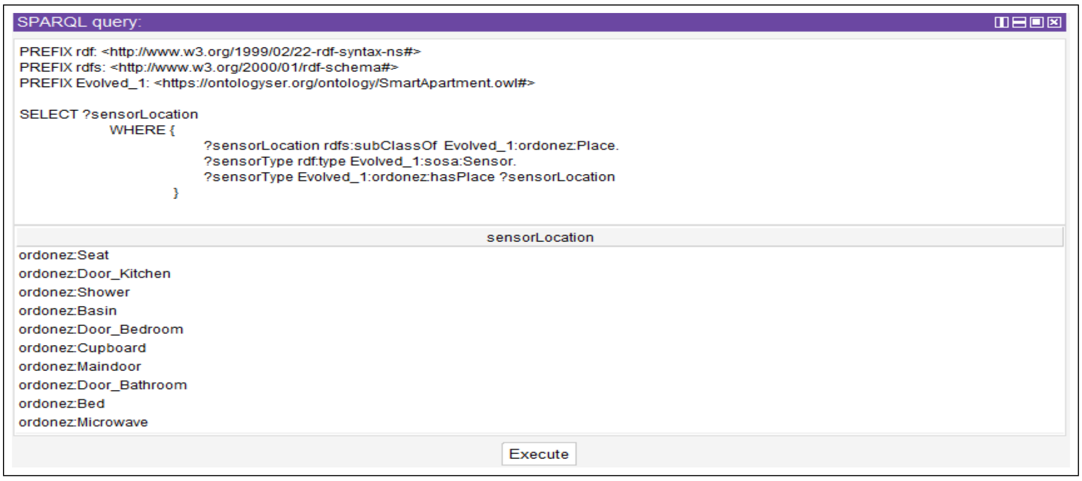

Figure 7.12: SPARQL query results for CQ1.2 at t_2.

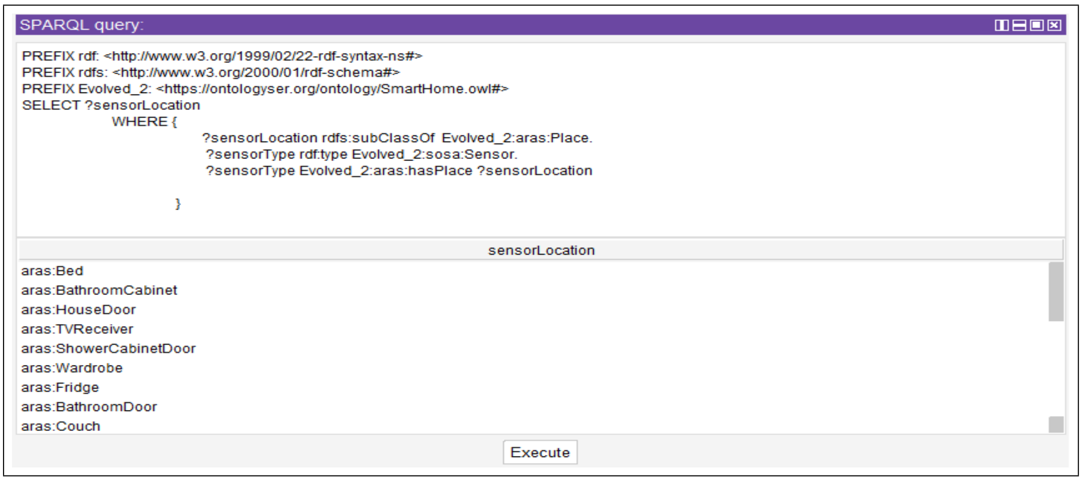

Figure 7.13: SPARQL query results for CQ1.2 at t_3.

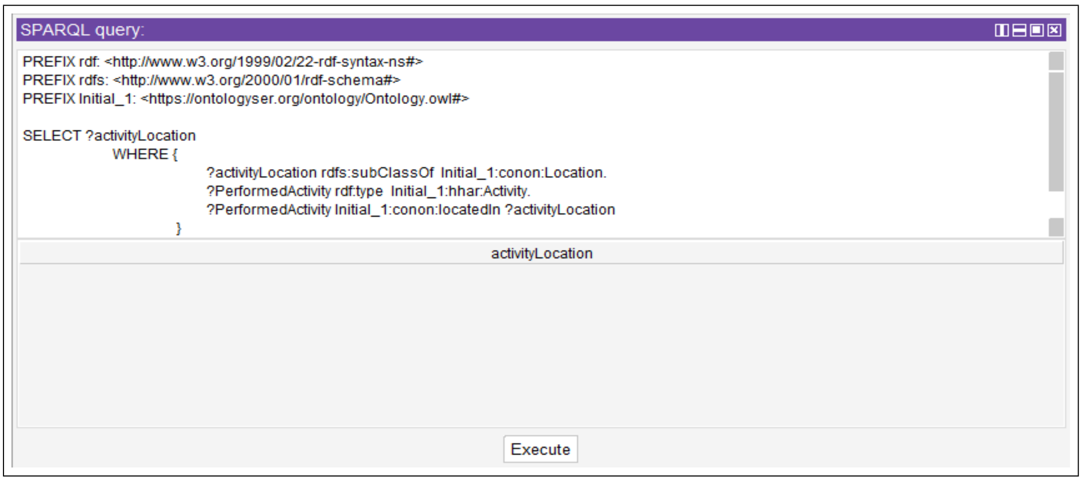

Figure 7.14: SPARQL query results for CQ2.2 at t_1.

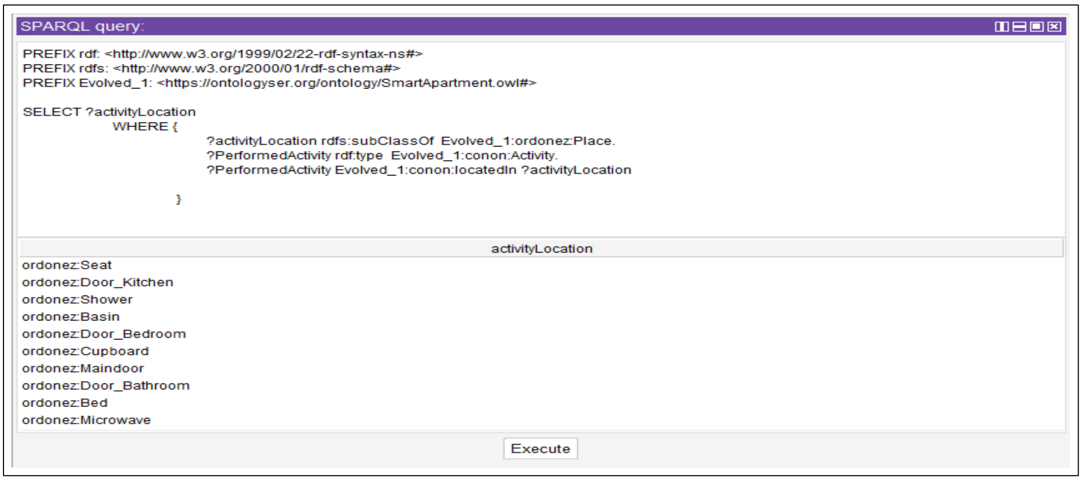

Figure 7.15: SPARQL query results for CQ2.2 at t_2.

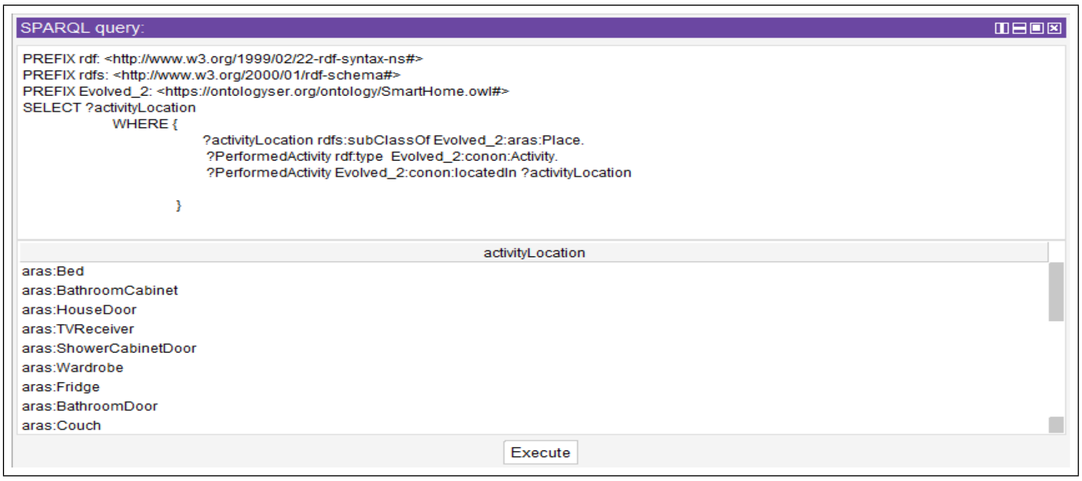

Figure 7.16: SPARQL query results for CQ2.2 at t_3.

	Ontology output results			
		Scenario 2		Scenario 3
	Initial at t_1	**Evolved at t_2**	**Initial at t_2**	**Evolved at t_3**
CQ1.1	hhar:Accelerometer, hhar:Gyroscope	ordonez:Electric, ordonez:Flush, ordonez:Magnetic, ordonez:PIR, ordonez:Pressure	ordonez:Electric, ordonez:Flush, ordonez:Magnetic, ordonez:PIR, ordonez:Pressure	aras:ContactSensor, aras:Distance, aras:ForceSensor, aras:IR, aras:Photocell, aras:SonarDistance, aras:Temperature
CQ2.1	hhar:upstairs, hhar:stand, hhar:walk, hhar:downstairs, hhar:sit	ordonez:Dinner, ordonez:Breakfast, ordonez:Grooming, ordonez:Leaving, ordonez:Lunch, ordonez:Showering, ordonez:Sleeping, ordonez:Snack, ordonez:Spare_Time TV, ordonez:Toileting	ordonez:Dinner, ordonez:Breakfast, ordonez:Grooming, ordonez:Leaving, ordonez:Lunch, ordonez:Showering, ordonez:Sleeping, ordonez:Snack, ordonez:Spare_Time TV, ordonez:Toileting	aras:WatchingTV, aras:WashingDishes, aras:UsingInternet, aras:Toileting, aras:TalkingOnThePhone, aras:Studying, aras:Sleeping, aras:Shaving, aras:ReadingBook, aras:PreparingLunch, aras:PreparingDinner, aras:PreparingBreakfast, aras:Other, aras:Napping, aras:ListeningToMusic, aras:Laundry, aras:HavingSnack, aras:HavingShower, aras:HavingLunch, aras:HavingGuest, aras:HavingDinner, aras:HavingConversation, aras:HavingBreakfast, aras:GoingOut, aras:Cleaning, aras:ChangingClothes, aras:BrushingTeeth
CQ1.2	hhar:Smartphone	ordonez:Seat, ordonez:Door$_K itchen, ordonez$: $Shower, ordonez$: $Basin, ordonez$: $Door_B edroom, ordonez$: $Cupboard, ordonez$: $Maindoor, ordonez$: $Door_B athroom, ordonez$: $Bed, ordonez$: $Microwave$	ordonez:Seat, ordonez:Door$_K itchen, ordonez$: $Shower, ordonez$: $Basin, ordonez$: $Door_B edroom, ordonez$: $Cupboard, ordonez$: $Maindoor, ordonez$: $Door_B athroom, ordonez$: $Bed, ordonez$: $Microwave$	aras:Bed, aras:BathroomCabinet, aras:HouseDoor, aras:TVReceiver, aras:ShowerCabinetDoor, aras:Wardrobe, aras:Fridge, aras:BathroomDoor, aras:Couch, aras:KitchenDrawer, aras:WaterCloset, aras:Hall, aras:Tap, aras:Chair, aras:Bed, aras:BathroomCabinet, aras:HouseDoor, aras:TVReceiver, aras:ShowerCabinetDoor, aras:Wardrobe, aras:Fridge, aras:BathroomDoor, aras:Couch, aras:KitchenDrawer, aras:WaterCloset, aras:Hall, aras:Tap, aras:Chair
CQ2.2	-	ordonez:Seat, ordonez:Door$_K itchen, ordon$ $Shower, ordonez$: $Basin, ordonez$: $Door_B edroom, ordonez$: $Cupboard, ordonez$: $Maindoor, ordonez$: $Door_B athroom, ordonez$: $Bed, ordonez$: $Microwave$	ordonez:Seat, ordonez:Door$_K itchen, ordon$ $Shower, ordonez$: $Basin, ordonez$: $Door_B edroom, ordonez$: $Cupboard, ordonez$: $Maindoor, ordonez$: $Door_B athroom, ordonez$: $Bed, ordonez$: $Microwave$	aras:Bed, aras:BathroomCabinet, aras:HouseDoor, aras:TVReceiver, aras:ShowerCabinetDoor, aras:Wardrobe, aras:Fridge, aras:BathroomDoor, aras:Couch, aras:KitchenDrawer, aras:WaterCloset, aras:Hall, aras:Tap, aras:Chair, aras:Bed, aras:BathroomCabinet, aras:HouseDoor, aras:TVReceiver, aras:ShowerCabinetDoor, aras:Wardrobe, aras:Fridge, aras:BathroomDoor, aras:Couch, aras:KitchenDrawer, aras:WaterCloset, aras:Hall, aras:Tap, aras:Chair

Table 7.5: Ontology based-context models vs. CQs.

From Figures 7.5, 7.8, 7.6, 7.9, 7.7, 7.10, 7.11, 7.12, 7.13, 7.14, 7.15 and 7.16, we can observe the following points:

- The CQs answers are complete for evolved ontology-based context models in the sense that they can provide correct and complete answers to all questions and then can serve their intended purposes.
- Semi-complete or partial answers are especially seen in queries for initial ontology-based context models that did not provide complete knowledge. The results from these CQs are not complete and show that knowledge in initial ontology-based context models is not complete and requires continuous evolution.
- There is a large difference between initial ontology-based context models and evolved ontology-based context model answers. The evolved ontology models in all cases provide more complete answers compared to the initial ontology-based context models. They could contribute more to the completeness.
- Comparison of the evolved ontology-based context models, achieved by the CoE approach, indicates that answers in most cases are better or even the same than the initial ontology-based context models. This is because of the nature of the evolution process, which aims to provide more extensive and complete knowledge.
- Last but not least, by analyzing evolved ontology-based context models with the set of CQs, we observed that applying evolution, in most cases, can provide more complete answers and less partial answers.

Summing up, we observe that the ability of evolved ontology-based context models to give correct answers to CQs. Also, we found that the answers to the CQs are satisfactory with our evolved ontology-based context models. Thus, ontology-based context models updated by utilizing a set of evolutions could provide more complete answers to the CQs in dynamic environments. This shows the feasibility and reliability of our CoE approach.

7.2.5 CoE Evaluation Discussion

In this subsection, we discuss the main obtained results after completing the CoE evaluation process. For simplifying the discussion of the results, they are explained in the following according to each applied evaluation approach.

From the feature-based evaluation perspective, the schema metrics' results, particularly for the RR and IRR, reflected a relatively minimal amount of non-taxonomic relations in the evolved ontology-based context models due to incompleteness in terms of the non-taxonomic relations and their inverses. In addition, we can infer that the IR is high indicating that the evolved ontology-based context model covers a wide range of concepts and notably captures the subsumption hierarchies. Therefore, the IR result underlined a vertical hierarchy in the evolved models with a large number of inheritance levels. In accordance with the IR result, the graph metrics emphasized the verticality of the evolved ontology-based models and guaranteed the structure's effectiveness. At the end, the obtained feature-based evaluation results could ensure the structure and schema quality of the evolved ontology-based context models.

From the criteria-based evaluation perspective, the results showed that evolved ontology-based context models could reach good consistency and conciseness. Conversely, the requirement for completeness was not met as well, since the minor pitfall P13, which deals with the issue of missing explicit declaration of inverse relationships, is pointed out. Thus, the evolved ontology-based models did not explicitly represent inverse relationships. Despite the noted issue, the obtained criteria-based evaluation results could confirm appropriate content quality for the given evolved ontology-based models since OOPS! conclusions attained by evolved context models were without any critical pitfalls. This showed that the CoE approach could be a potential candidate to evolve an existing ontology-based context model in an automatic way at runtime.

From the expert-based evaluation perspective, the results indicated that the evolved ontology-based models exceeded in terms of environment coverage, which was proven by the higher mean scores for precision and recall. The improvement in the recall score was higher than that for precision, which reveals that the evolved ontology-based models were better in coverage against the initial ontology-based models. Therefore, the coverage requirement of the evolved ontology-based models was considered acceptable, since the evolved ontology-based models had the advantage of producing high precision and good recall. Despite this, there was room for improvement regarding the coverage requirement through providing minor enrichments, since the score of the F1-score led to around 0.72. Therefore, invited experts stressed, in an attempt to achieve a higher coverage, the need for slight enrichments of the evolved ontology-based models with additional knowledge to foster the experts' coverage agreement.

And finally, from the competency question-based evaluation perspective, the results provided evidence that the evolved ontology-based context models' answers to the CQs can reach a complete set of answers. Thus, there is a large difference between initial and evolved ontology-based context models' answers. The ontology models evolved by utilizing the CoE approach could provide more complete answers to the CQs compared to the initial models. This is because of the nature of the proposed evolution approach that demonstrates its feasibility and reliability.

To conclude, the overall results were promising and largely showed an appropriate quality of content, schema and structure in the evolved ontology-based models. They reflected a considerable consistency, conciseness and coverage, whereas the completeness was not met that well. By achieving both validation and verification, we can confirm the quality of the evolved ontology-based models for answering changes arising in the surrounding environments at runtime and achieving the purpose of the CoE approach.

7.3 DMA Evaluation

In this section, we concentrate on evaluating the decision-making adaptation approach presented in chapter 6. This experimental evaluation discusses the effectiveness of the proposed DMA approach. For this, a range of experiments were carried out to investigate the effectiveness in terms of number of rules, performance and computational time.

7.3.1 Experimental Setup

All experiments were conducted on six benchmark data sources of varying complexity. The name, number of instances, number of attributes and number of class labels in each data source acquired from the UCI Machine Learning Repository are included in Table 7.6.

Data source	Number of instances	Number of attributes	Number of classes
Breast	286	10	2
HHAR	43,930,250	16	26
Ordonez	20,358	24	37
ARAS	5,184,000	80	68
Iris	150	4	3
Adult	48842	15	2

Table 7.6: Characteristics of data sources used in experiments.

For evaluation purposes, we utilized standard open-source implementations of Machine Learning algorithms and rule mining algorithms provided by Weka [Witten et al., 1999] in a 10-fold cross-validation evaluation protocol in order to get accurate results for all data sources. In 10-fold cross-validation protocol, the entire benchmark data sources are partitioned into ten parts of equal size and nine parts of them are used at a time for training and the remaining one is used for testing. The process is repeated ten times, with different partitions used as training data and test data. In addition, we set the minimum expected weighted confidence threshold to 1 and support threshold to 0.5 in order to generate the results described in the following experiments. Furthermore, the various GA parameters were selected. Crossover and mutation probabilities were taken respectively as 0.5 and 0.01. The size of the initial population depends on the number of the generated rules from the benchmark data sources. Thus, the initial population size ranges from 14 to 439 and the maximum number of generations is set to 100.

7.3.2 Experimental Metrics

For analyzing the effectiveness of the proposed DMA approach, some well-known metrics were used. Therefore, the following parameters were considered for these metrics:

- True Positive (TP) is the number of rules positively predicted that is actually positive;
- True Negative (TN) is the number of rules negatively predicted that is actually negative;
- False Positive (FP) is the number of rules positively predicted that is actually negative;
- False Negative (FN) is the number of rules negatively predicted that is actually positive.

Based on the previous parameters, the following metrics are:

- Precision that reflects the ability of an approach to return relevant rules among a set of irrelevant and relevant rules. The precision can be computed by Equation 7.4.

$$Precision = \frac{TP}{TP + FN} \tag{7.4}$$

- Recall that reflects the ability of an approach to return relevant rules only. The Recall is defined as given in Equation 7.5.

$$Recall = \frac{TP}{TP + FP} \tag{7.5}$$

- Accuracy that reflects ability of an approach to return the accurate rules over the all rules made in the data source as can be seen in Equation 7.6.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{7.6}$$

Apart from these metrics, we considered the computational time that reflects the average time required for generating the set of IF-THEN decision rules in terms of seconds.

7.3.3 Effectiveness Analysis

7.3.3.1 Rule Analysis

In this experiment, the number of decision rules generated by the DMA approach and the traditional algorithms of rule mining, such as Apriori and FP-Growth, was evaluated. Figures 7.17, 7.18, 7.19, 7.20, 7.21 and 7.22 illustrate the number of generated rules for the benchmark data sources.

Observing Figures 7.17, 7.18, 7.19, 7.20, 7.21 and 7.22, we can see that the traditional algorithms generate the highest number of rules, while the DMA approach generates the lowest number of rules on all data sources. In particular, the DMA approach generates 4117 decision rules on average against 6 data sources, whereas the Apriori and FP-Growth algorithms derive 9614 and 11045 on average, respectively. Thus, the results show that the number of generated decision rules using traditional algorithms is large and huge. The reason beyond these results is that traditional algorithms simply take into account all combinations of attributes while generating rules. In contrast, the results indicate that the number of generated rules by our DMA approach is small. The downtrend reveals that our DMA approach could generate the minimum number of decision rules comparing the Apriori and FP-Growth and could keep the number of discovered rules as small as possible. As a result, for a high confidence value, traditional algorithms satisfy significantly more rules than the proposed DMA approach. Therefore, the DMA approach could generate a reasonably smaller number of decision rules compared with traditional rule mining algorithms since it abandons the redundant rules and retains the non-redundant rules.

7.3.3.2 Performance Analysis

In this experiment, we discussed the effectiveness of the DMA approach in terms of performance measures such as Precision, Recall and Accuracy. For this purpose, we compare the performance of the proposed approach with well-known Machine Learning algorithms, namely NB, JRip and Decision Table (DT). The reason for selecting these algorithms is that they generate rule-based classifiers and have high performance compared with other algorithms [Rubini and Eswaran, 2015]. We also compare the performance of the

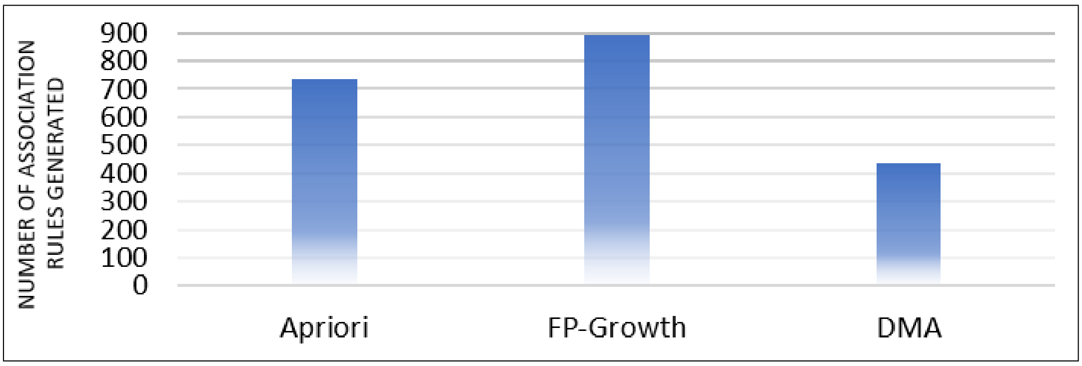

Figure 7.17: Rule analysis for the Breast data source.

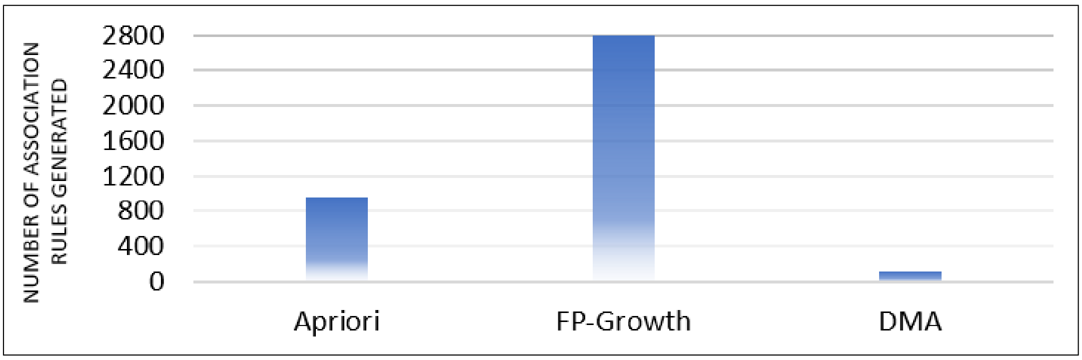

Figure 7.18: Rule analysis for the HHAR data source.

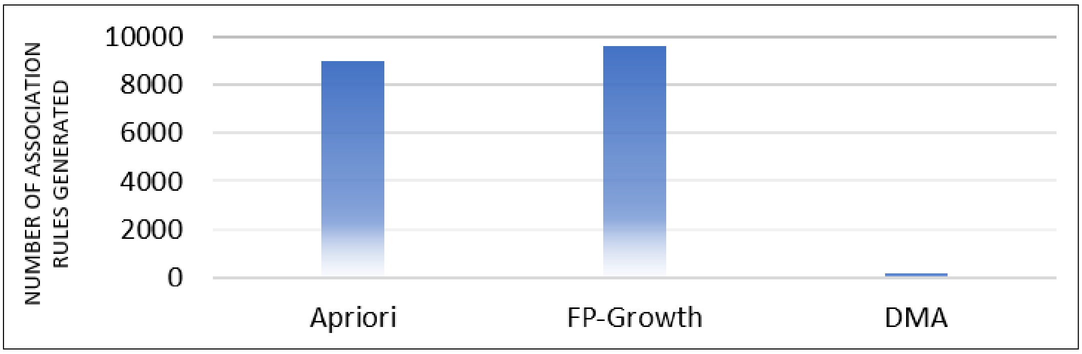

Figure 7.19: Rule analysis for the Ordonez data source.

DMA approach with some similar state-of-the-art works, including [Hong et al., 2009], [Sarker, 2019] and [Basha, 2021]. Experimental results on performance measures are highlighted in Figures 7.23, 7.24, 7.25, 7.26, 7.27 and 7.28.

Our observations on candidate data sources show that the DMA approach consistently outperforms the compared Machine Learning algorithms for generating decision rules by maximizing the precision and recall. In addition, our observations reveal that the DMA approach achieved better accuracy than NB, JRip and DT on HHAR, Ordonez, ARAS and Iris data sources. For instance, the accuracy for NB, JRip and DT on the Iris data source are 95.33%, 96%, 95.3% and 92.66%, respectively, whereas for the proposed DMA approach the accuracy is 98%. Moreover, the obtained results confirm that the DMA approach has outstanding performance compared with state-of-the-art works. For instance, we achieve

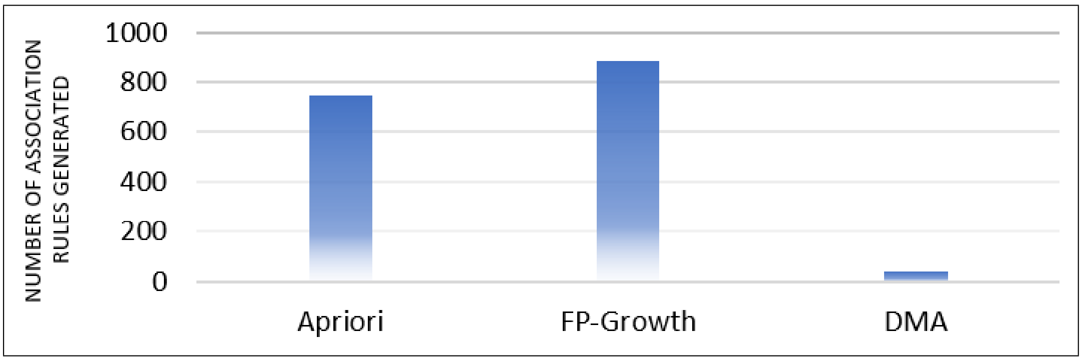

Figure 7.20: Rule analysis for the ARAS data source.

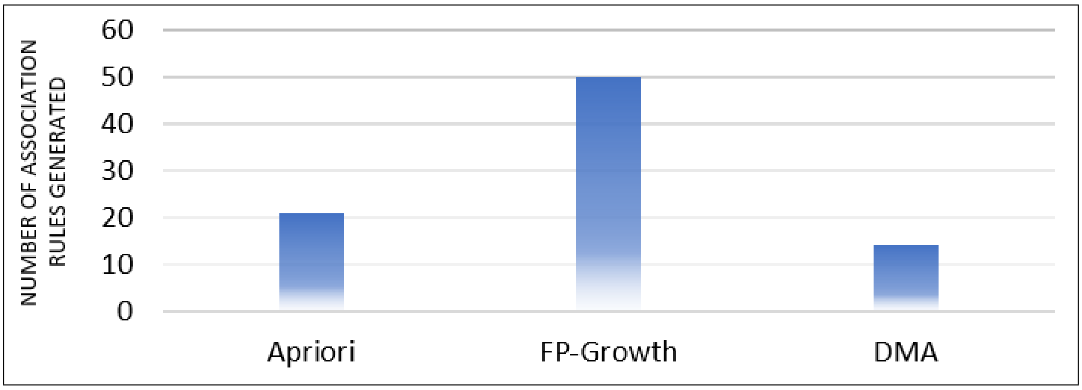

Figure 7.21: Rule analysis for the Iris data source.

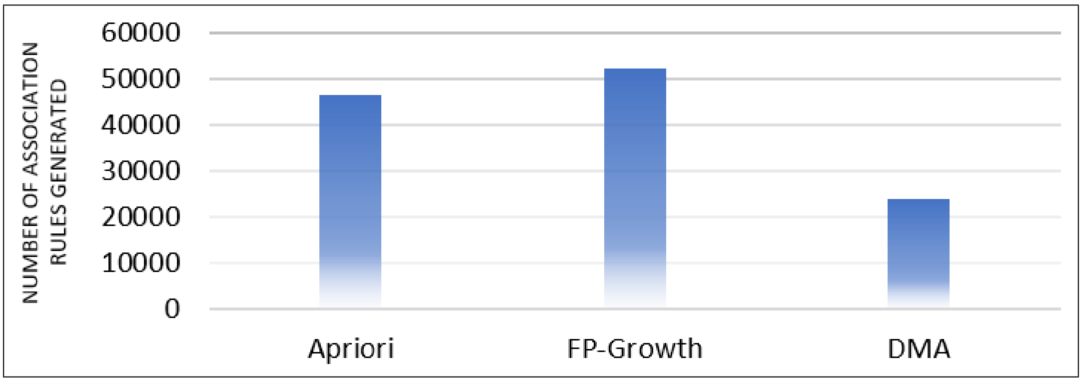

Figure 7.22: Rule analysis for the Adult data source.

an approximately 2% and 3.2% accuracy gain compared with works of [Hong et al., 2009], [Sarker, 2019] and [Basha, 2021] when dealing with Breast and Adult data sources, respectively. Thus, obtained results proved that the proposed DMA approach tends to get reasonably high accuracy on all data sources. Therefore, we can conclude that the DMA approach is more effective relative to the compared Machine Learning algorithms and state-of-the-art works while generating decision rules since we capture decision rules from both more performant Machine Learning algorithms that lead to improve the performance results.

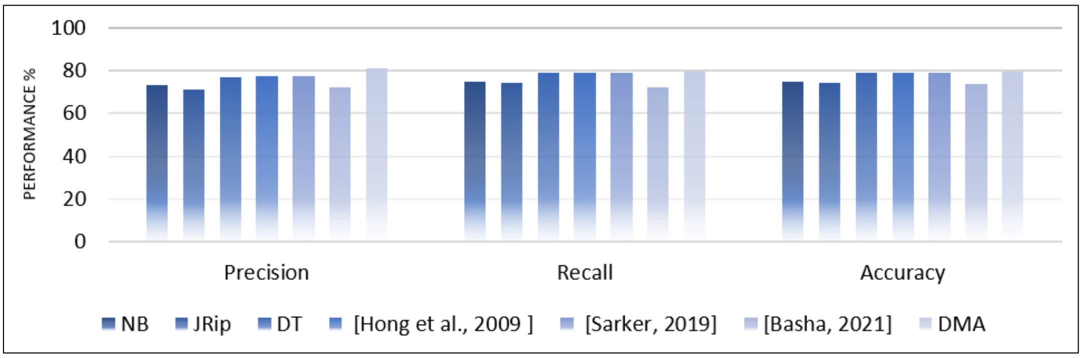

Figure 7.23: Performance analysis for the Breast data source.

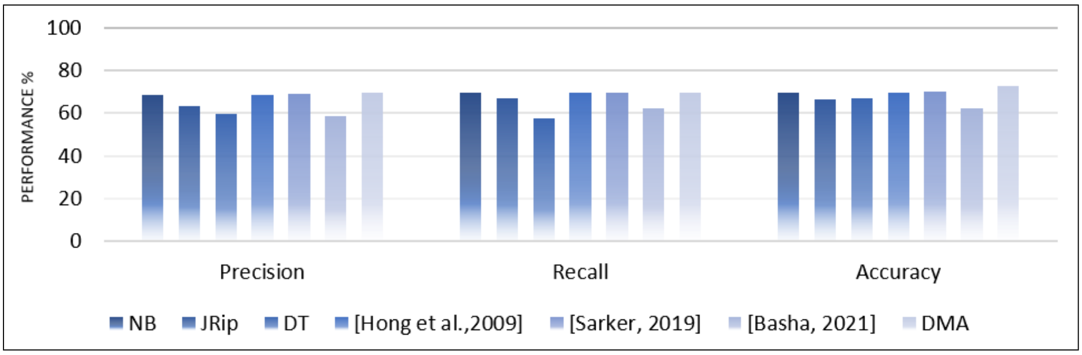

Figure 7.24: Performance analysis for the HHAR data source.

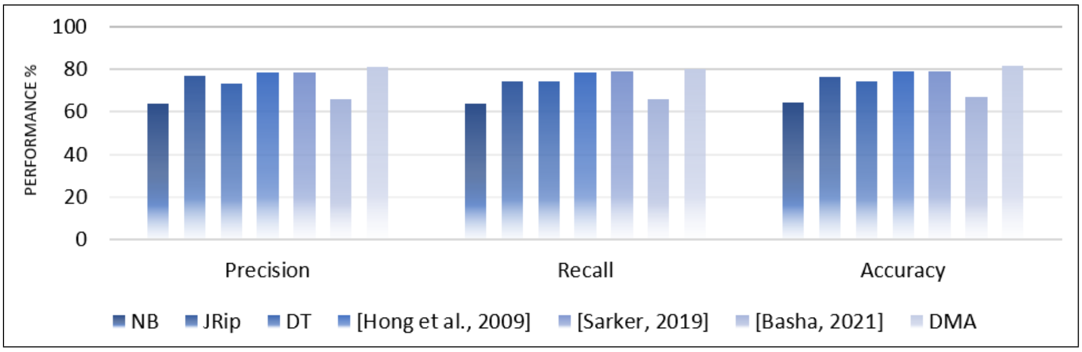

Figure 7.25: Performance analysis for the Ordonez data source.

7.3.3.3 Computational Time Analysis

In this experiment, we compared and analyzed the computational time of the DMA approach, the Machine Learning algorithms mentioned earlier and the state-of-the-art works [Hong et al., 2009, Sarker, 2019]. In this experiment, we did not consider the work presented by [Basha, 2021] since its GA is not publicly available. To this end, the sizes of the data sources were fixed since the computational time may vary based on the data source size. Table 7.7 illustrates the time consumed by the DMA approach for generating decision rules against the selected Machine Learning algorithms and the state-of-the-art works in all data sources.

From the illustrated results, the average time spent on each data source to generate a set of decision rules is 0.38 seconds in the DMA approach. The experimental results clearly

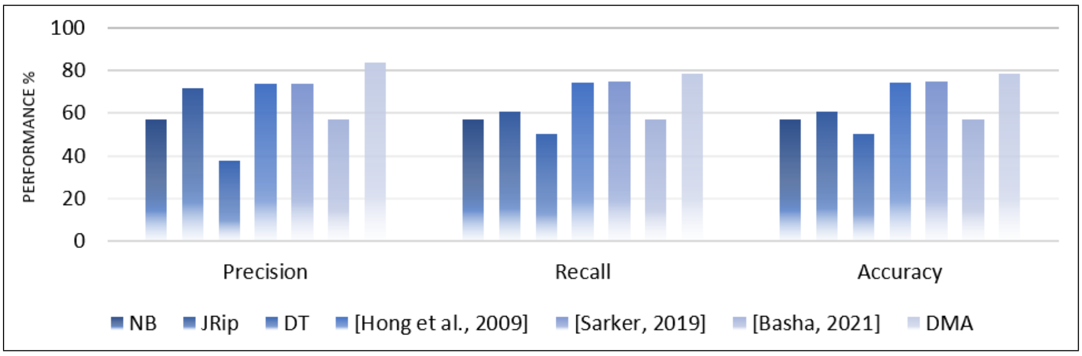

Figure 7.26: Performance analysis for the ARAS data source.

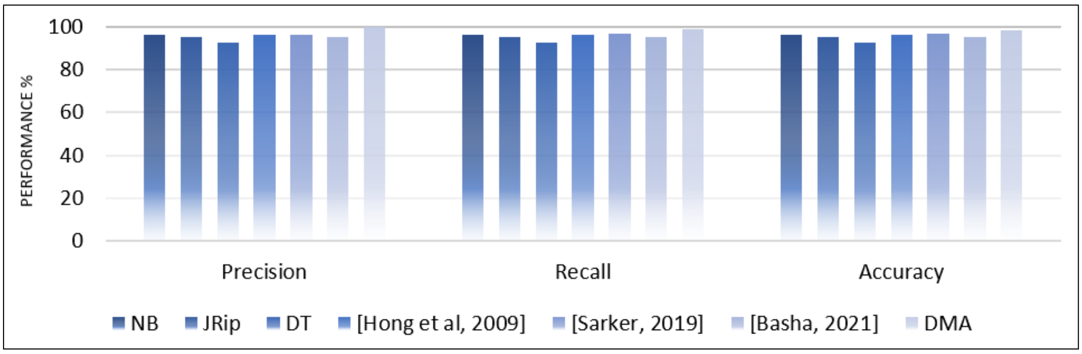

Figure 7.27: Performance analysis for the Iris data source.

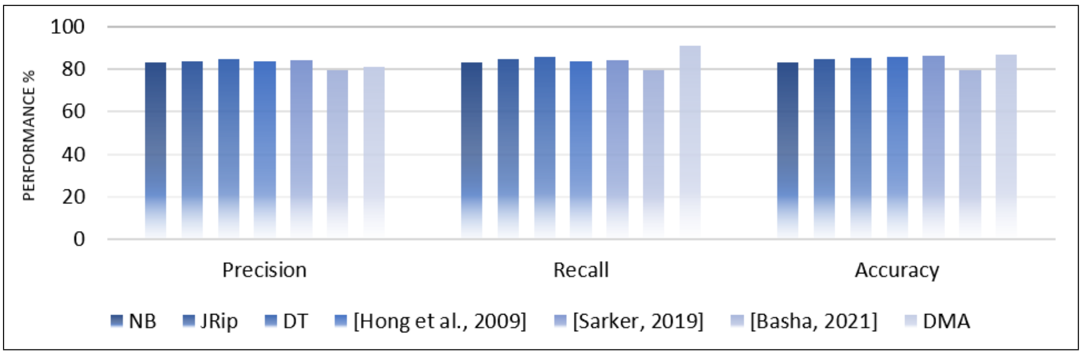

Figure 7.28: Performance analysis for the Adult data source.

show that the computational time of the DMA approach is slightly increased compared with the selected Machine Learning algorithms and with the state-of-the-art works, but the computational time on the whole is not much different. The slight increase can be explained by the fact that we apply the GA for optimization on the basis of both Machine Learning algorithms, which could slightly increase the time complexity of the rule generation process compared with the selected Machine Learning algorithms.

7.3.4 DMA Discussion

We discuss the encouraging obtained results of the proposed DMA approach. Firstly, the DMA approach outperformed compared to the well-known traditional rule mining algorithms in terms of the number of rules by eliminating the redundant generation for each data source. It generated, on average far, fewer rules than those generated by the

Data source	NB	JRip	DT	[Hong et al., 2009]	[Sarker, 2019]	DMA
Breast	0.05	0.04	0.02	0.03	0.05	0.06
HHAR	0.02	0.02	0.05	0.04	0.06	0.16
Ordonez	0.02	0.05	0.06	0.04	0.06	0.10
ARAS	0.05	0.04	0.03	0.03	0.05	0.08
Iris	0.02	0.01	0.02	0.02	0.04	0.06
Adult	1.13	1.48	1.94	1.08	1.10	1.82
Average	0.29	0.27	0.35	0.21	0.23	0.38

Table 7.7: Computational time analysis for the benchmark data sources in seconds.

traditional algorithms included in the comparison. Thus, it provided a reasonably smaller number of rules on smaller and bigger data sources compared to the traditional algorithms. This is due to the fact that the DMA approach is based on hybrid Machine Learning that takes advantage of Machine Learning to extract the relevant relationships and to eliminate the redundancy while generating decision rules. In brief, the DMA approach significantly reduces the total number of generated rules and outputs a well-performed set of decision rules using the extended GA. Such non-redundant and well-performed decision rule generation makes the proposed DMA approach more effective and can be used to provide an automatic decision-making adaptation regarding changes occurring in users' surrounding environments at runtime. Secondly, the DMA approach exceeded the well-known selected Machine Learning algorithms as well as certain state-of-the-art works in terms of precision, recall and accuracy. In particular, it achieved the best accuracy on HHAR, Ordonez, ARAS and Iris data sources among all selected Machine Learning algorithms. However, it got slightly worse results in terms of accuracy on Breast and Adult data sources since imbalanced data sources may lead to slightly worse accuracy results. Even though not achieving the best accuracy results on these data sources, the DMA approach achieves almost reasonably high accuracy in most data sources. Moreover, we must stress that the main advantage of the DMA approach, in contrast to the compared state-of-art works, is its ability to achieve a good trade-off between precision, recall and accuracy on multiple data sources. Thirdly, the DMA approach could be somewhat more time-consuming than the selected Machine Learning algorithms as well as the compared state-of-the-art works. The slight increase in time is reasonable due to the hybridization and optimization. However, this loss of the little running efficiency will result in a significant improvement in the quality of the results in terms of number of rules, precision, recall and accuracy. Regarding the time-consuming results, it is noted that the effectiveness of the proposed DMA approach does have room for improvement, which is also the need for further improvement in the future work.

Overall, the findings of the experimental study reveal that the DMA approach can

1. effectively minimize the issues of redundant rule generation,

2. provide a promising performance and a high accuracy to extract a concise set of decision rules and

3. take slightly more time for generating a set of well-performed rules.

7.4 IConAS Approach Evaluation

The present section focuses on the evaluation of the IConAS approach introduced in chapter 4 through an Android mobile application example of the proposed IntElyCare framework. This section can be broken down into the following:

- First, subsection 7.4.1 provides details of a case study that we apply to the proposed IntElyCare framework in two application scenarios;
- Next, subsection 7.4.2 presents an evaluation study to demonstrate the effectiveness of our activity recognition in the proposed IntElyCare framework;
- Then, subsection 7.4.3 conducts a feasibility study to assess user satisfaction with the application.

7.4.1 Case Study

7.4.1.1 Case Study implementation

This subsection gives an introduction into the technological realization of the IntElyCare framework prototype. As previously mentioned, the healthcare domain is selected. The implementation of the IntElyCare framework can be separated into three main parts, the Frontend implementation (Client side) as well as the Backend and the Backend-For-Frontend (BFF) implementation (Server side). For that, an overview about the overall implementation architecture is given, which is followed by a short outline about the used technologies and programming languages for the prototypical realization. It is worth highlighting that we have omitted several implementation details where we thought they were not essential. Afterwards, the Frontend, the BFF and the Backend of the prototypical solution are each elaborated in detail. Figure 7.29 illustrates the overall IntElyCare implementation architecture. The focus of the illustrated architecture is on the communication of BFF microservices with Backend microservices and Frontend. The three parts of the architecture usually run-on distributed client and server, which communicate using REST as a communication basis.

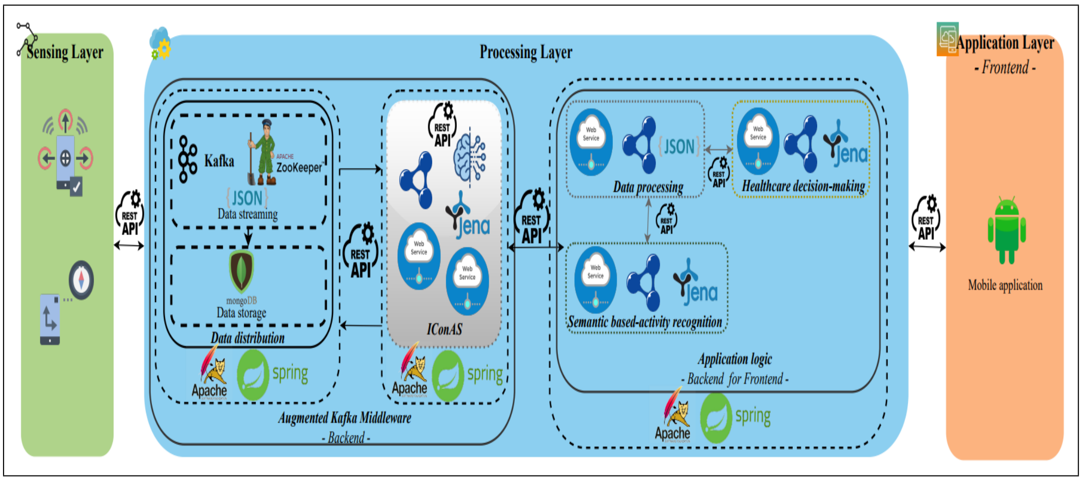

Figure 7.29: IntElyCare framework implementation architecture overview.

As depicted in Figure 7.29, the Integrated Development Environment (IDE) used for Frontend development is Android Studio and Frontend framework is Android. As IDE for BFF and Backend development, Eclipse is used as well as Apache Spring, and more precisely, Spring Boot framework [Walls, 2015] is used as Backend framework for the microservice implementation. Spring is currently one of the most popular open-source development frameworks for Java-based application servers. The presented technological details for each single part of the architecture allow stating that the chosen technological support is not limited to the displayed ones. After introducing the architecture and the technological foundations for the prototype, the following subsections deals with the Backend, BFF and Frontend development details. At first, an overview for the whole realization of the Backend is given. Subsequently, the major realized views of the BFF and Frontend are presented, respectively.

1. **Backend**
   Due to the difficulties when trying to integrate Apache Kafka directly with Android, Kafka producers and consumers are meant to be used on the Backend. For this reason, the Backend has been constructed with Spring Boot. With it, REST endpoints were exposed that can be called directly from our BFF and our mobile application. Messages are published from the elderly's device to the Backend and are later handled by the Kafka producers. In order to construct our Backend, four Kafka interfaces have been used:

   - Kafka producer API allows applications to send output streams of data to topics in the Kafka cluster;
   - Kafka consumer API allows applications to read input streams of data from topics in the Kafka cluster. The consumers ingest data generated by the producers by subscribing to a particular topic;
   - Kafka streams API allows transforming streams of data from input topics to output topics;
   - Kafka connector API allows you to build and run reusable producers or consumers to connect Kafka topics to existing applications or databases.

   Both Kafka consumers and producers communicate with schema registry to either register the schema, in case of producers, or obtain schema, in case of consumers, for a particular topic. A topic is created in the Apache Zookeeper first, and then persisted into the MongoDB database. Storing data at two or more different locations comes with its own challenges, including but not limited to, synchronization problems and handling conflicts. Detailed topic information is retrieved from the Zookeeper using the Kafka consumer API. Therefore, Kafka uses Zookeeper to keep information about topics. In addition to the Kafka implementation, our proposed IConAS approach, which augments the Kafka middleware, is implemented with Spring Boot framework and microservice architecture. The whole IConAS approach consists of two individual microservices related to the CoE and DMA approaches. All of the microservices are structured into the Processing layer and typically communicate over REST APIs. HTTP requests are delegated from the BFF to the Backend parts. Additional implementation details were described in chapter 5 and chapter 6.

2. **BFF**
   As outlined by [Newman, 2021], BFF is a pattern in which use is increasingly related to Frontends. It is characterized as being an API placed on the server-side, that will contain all the logic that once would be in the backend of the Web or mobile application. The idea of the BFF part is that for every kind of the Frontend (e.g., web, desktop, mobile application, etc.) a corresponding server-side component, called BFF, is customized. It intends to strip all the application logic associated with the Frontend. In our case, the BFF bridges the gap between the requirements of the Frontend to the Backend. It aggregates multiple microservices that correspond to a single Frontend. As shown in Figure 7.29, the BFF is part of the server-side and serves as a communication middle layer between the Frontend and various available Backend microservices. It is developed using the Spring Boot framework. One of its purposes is to receive requests from the Frontend through its REST endpoint and sending the data to the Backend through the Kafka APIs.

3. **Frontend**
   The Frontend, also known as the client-side, represents the User Interface (UI) and what an elderly can interact with and see on the mobile application. It is a native Android application. Because of our BFF architecture, our framework can handle any type of Frontend. Despite that, we chose to develop an Android-based Java application. When deciding which programming language to use in the development of the Android application, we had several options. We ultimately decided on Java, because of the amount of online support and resources available, whereas Kotlin is a relatively new language, with a smaller community, which means it may be harder to find solutions to common problems.

7.4.1.2 Case Study Application

In order to provide a clear idea of how the implemented mobile-based application works and has the potential to accurately infer performed activities, current situations and to determine relevant services for elderly, a case study is drawn out depicting changes in users' preferences and behaviors at runtime. The objective of this case study is to explore the application of the proposed CoE and DMA in Healthcare domain. The present case study concerns an elderly, named David, who downloads the application of the IntEly-Care framework to his mobile. This latter contains common motion sensors such as an accelerometer, so that inertial data can be continuously collected at runtime. In this case study, two application scenarios, occurring in 8 and 6 stages, respectively, are adopted and illustrated in Figures 7.30 and 7.35, where blue icons indicate known activities and red icons indicate unknown activities that will be introduced as new activities to the initial activity ontology module. The details of these two scenarios are described below.

1. **Scenario 1: HHAR data source**

   Figure 7.30 shows the first scenario that uses an initial activity ontology module including only certain activities, such as sitting and standing, and evolved activity ontology modules that includes new activities, such as walking, walking upstairs and walking downstairs, in addition to the initial ones. The first scenario runs as follows:

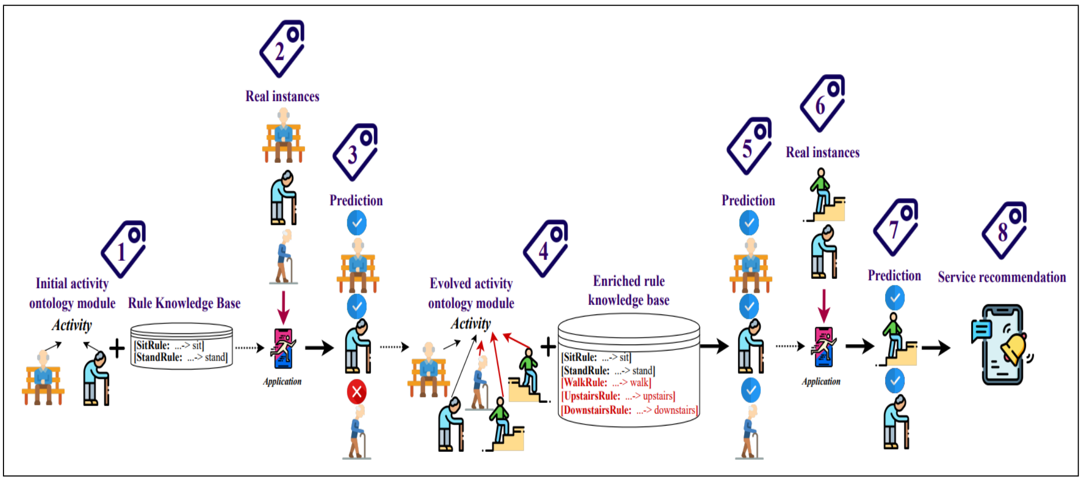

Figure 7.30: A first scenario for an elderly with a random activity set in the initial activity ontology module.

a) **Stage 1.** At this stage, the application is only able to recognize a set of two common predefined activities, namely sitting and standing, modeled on the initial activity ontology module as shown in Figure 7.31, previously illustrated as Figure 4.10 in chapter 4;

b) **Stage 2.** The application tracks David movements when performing the two common predefined activities that are already supported by the initial activity ontology module. In addition, David performs a walk activity, which is not one of the activities recognized by the application;

c) **Stage 3.** Then, the application automatically recognizes sitting and standing activities, while a misclassification occurs when David carries out the walking activity that is recognized as "unknown" activity as illustrated in Figure 7.32(a);

d) **Stage 4.** To reinforce activity recognition capability and learn new activities, the application evolves the initial activity ontology module with new activities discovered from HHAR data source, which is publicly available [Stisen et al., 2015]. This data source contains data originating from an accelerometer sensor and is labeled with 5 labels according to specific activities, such as "sit", "stand", "walk", "upstairs" and "downstairs". More specifically after the ontology-based context evolution using the CoE approach, activity concepts, such as "walk", "upstairs" and "downstairs", are involved in the initial activity ontology module as shown in Figure 7.33. It should be pointed out that the ontology-based context model evolution process is quite similar to the process in scenario 1 presented in subsubsection 5.4.2.1. Following this evolution, a decision rule generation is performed using the DMA approach to enrich the existing rule knowledge base with new rules corresponding to the new activities as depicted in Figure 7.34. It is noteworthy that the same rule generation process as the process detailed in subsubsection 6.4.2.1, is carried out, where the HHAR data source is used;

e) **Stage 5.** Once again, the application accurately recognizes the unknown activity using the evolved activity ontology module and the enriched rule knowledge base as shown in Figure 7.32(b);

f) **Stage 6.** After a while, the application tracks David movements, for the second time, when performing some activities that are already supported by the evolved activity ontology module;

g) **Stage 7.** Once again, the application automatically recognizes the performed activities "walk upstairs" and "walk" as depicted in Figure 7.32(c);

h) **Stage 8.** After a while of walking, David needs a place to rest as he has an unhealthy health status and suffers from coronary heart disease. When walking for a long period, he feels a pressure close to his heart and he is short of breath. Therefore, this situation provides a safety need and David's application triggers a notification for the health recommendation service under the elderly's safety needs-related service category. The recommended safety needs-related service is performed to remind David that he must adhere to bed rest or sit activity. An example of the triggered notification is presented in Figure 7.32(d).

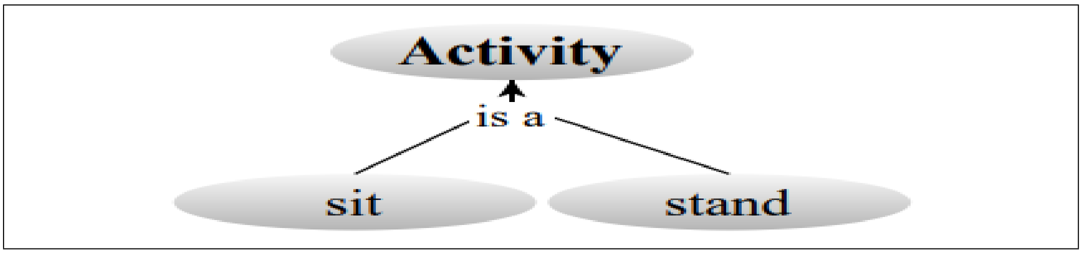

Figure 7.31: An example of a closed set of predefined activities in the Activity ontology module.

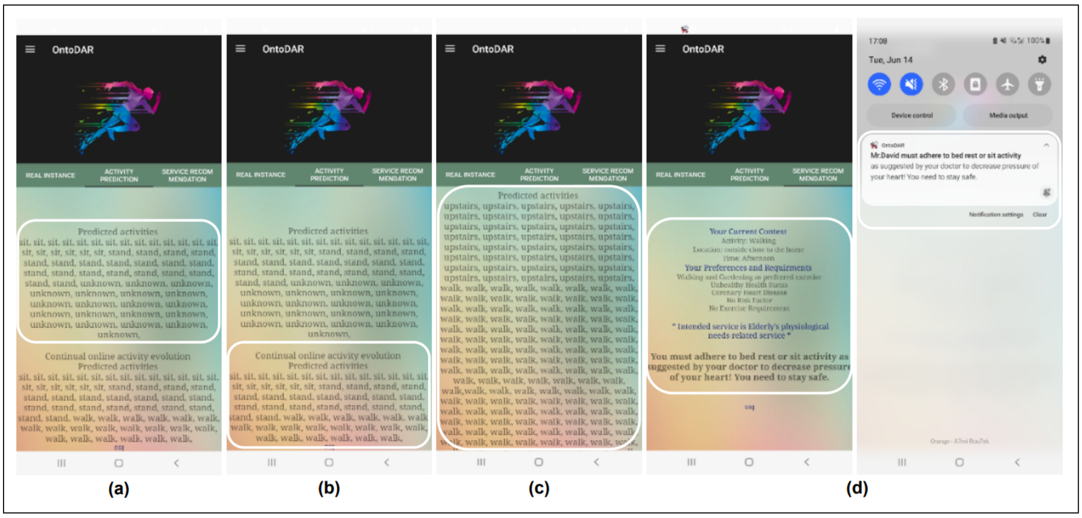

Figure 7.32: Activity recognition results of (a) stage 3; (b) stage 5; (c) stage 7 and (d) stage 8 using HHAR data source.

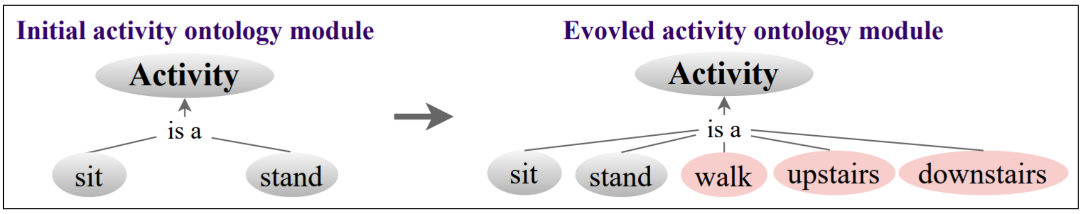

Figure 7.33: Evolved activity ontology module with new activities using HHAR data source.

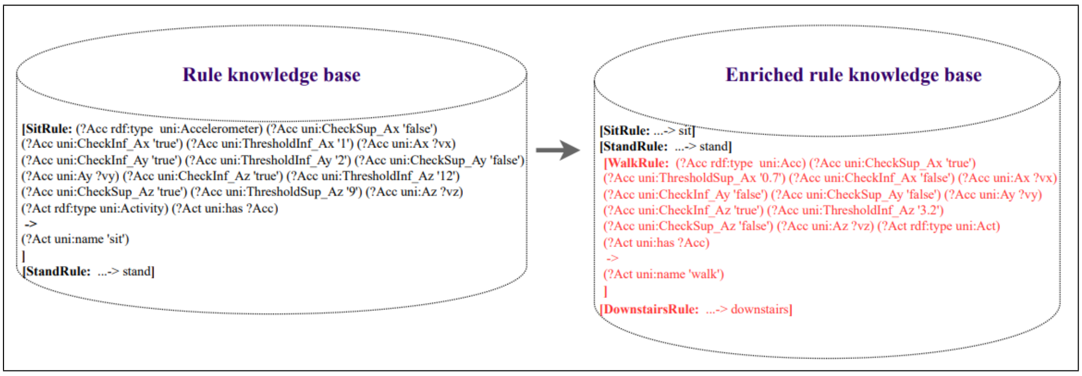

Figure 7.34: Enriched rule knowledge base using HHAR data source.

2. **Scenario 2: UCI-HAR data source** Figure 7.35 shows the second scenario that uses an initial activity ontology module containing specific activities, such as sitting, standing, walking, upstairs and downstairs, and an evolved activity ontology module that includes a new activity in addition to the initial ones. The second scenario runs as follows:

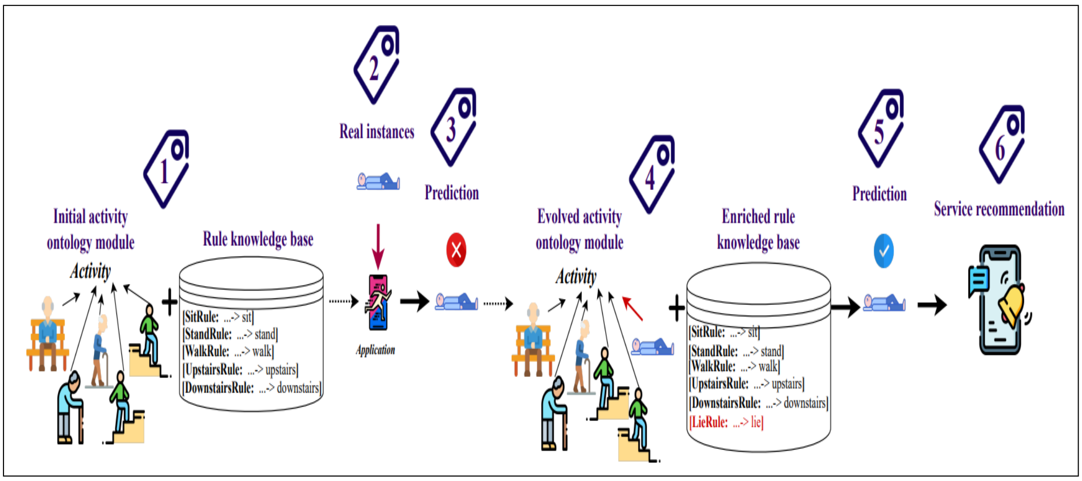

Figure 7.35: A second scenario for an elderly with a random activity set in the initial activity ontology module.

- **Stage 1.** At this stage, the application is able to recognize a set of common predefined activities, namely sitting, standing, walking, walking upstairs and walking downstairs, modeled on the initial activity ontology module that is previously shown in Figure 7.33;

- **Stage 2.** The application tracks David movements when performing lying activity as previously specified in the notification (see Figure 7.32(d)), where lying activity is not observed in the initial activity ontology module;

- **Stage 3.** Then, the application inaccurately recognizes lying activity when David was in the lying position that is recognized as "unknown" as depicted in Figure 7.36(a);

- **Stage 4.** To reinforce activity recognition capability and learn new activities, the application evolves the initial activity ontology module with new activities discovered from UCI HAR data source, which is publicly available [Anguita et al., 2013]. This data source contains data originating from an accelerometer sensor and is labeled with 6 labels according to specific activities, such as "sit", "stand", "lie", "walk", "walk upstairs" and "walk downstairs". More specifically after the ontology-based context evolution, the activity concept "lie" is involved in the previously evolved activity ontology module as shown in Figure 7.37. Following this evolution, a decision rule generation is performed to enrich the rule knowledge base with a new rule corresponding to the lying activity;

- **Stage 5.** Once again, the application accurately recognizes the unknown activity using the evolved activity ontology module and the enriched rule knowledge base as shown in Figure 7.36(b);

- **Stage 6.** After lying for some time, inactivity can worsen the disease. In this regard, the application often focuses on reducing levels of physical inactivity for David, impacting on adherence to physical activity, promoting all the benefits related to exercise. Therefore, this situation raises a physiological need and enables the application to trigger the exercise recommendation service under the category of elderly's physiological needs-related services. The triggered notification service is performed to convince David to get out and enjoy some gardening since he prefers walking and gardening as exercises. An example of the triggered notification is displayed in Figure 7.36(c).

7.4.2 Activity Recognition Evaluation

In this subsection, we provide experimental analysis to first assess the proposed IntElyCare framework in terms of activity recognition performance and then to demonstrate its advantage over some of existing works. In this experimental analysis, we chose the two publicly available data sources, the HHAR and UCI HAR data sources. Details on each data source was elaborated in subsubsection 7.4.1.2.

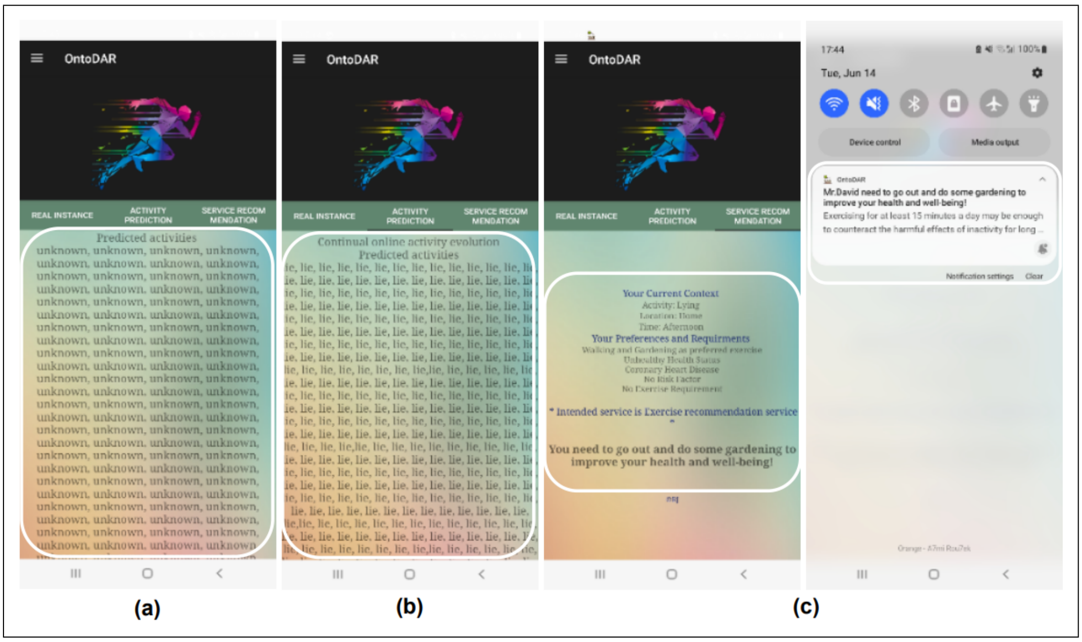

Figure 7.36: Activity recognition results of (a) stage 3, (b) stage 5 and (c) stage 6 using UCI HAR data source.

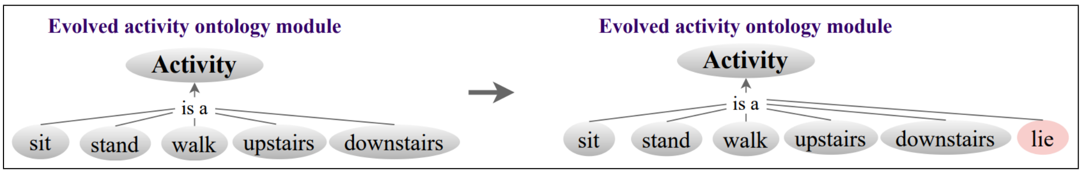

Figure 7.37: Evolved activity ontology module with new activities using UCI-HAR data source.

7.4.2.1 Performance Metrics

Multiple performance evaluation criteria are applied to assess the activity recognition performance. The accuracy measures the effectiveness of a predictive model by considering the correctly predicted instances to the ratio of the total number of instances. Since accuracy is not the best measure in the case of imbalanced data sources, we are using additional performance metrics to alleviate this limitation. Therefore, three more performance metrics namely precision, recall and F1-score are used as additional performance measures. Equations 7.7, 7.8, 7.9 and 7.10 presents the mathematical expressions of accuracy, precision, recall and F1-score, respectively which are calculated based on (TP), (FN), (FP) and (TN) instances on the collected inertial data [Gad et al., 2018].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{7.7}$$

$$Precision = \frac{TP}{TP + FN} \tag{7.8}$$

$$Recall = \frac{TP}{TP + FP} \tag{7.9}$$

$$F1 - score = 2\,\frac{Precision\ Recall}{Precision + Recall} \tag{7.10}$$

7.4.2.2 Activity Recognition Performance Evaluation on the HHAR Data Source

1. **Performance in Discovering Unknown Activities on the HHAR Data Source**

   To assess the performance in discovering unknown activities, we focus on evaluating the accuracy of discriminating unknown and known activities. In this experiment, we do not perform the Continual online activity evolution to see whether the discovery of unknown activities is effective. In this case, the IConAS approach including CoE and DMA approaches, is omitted. To validate this experiment, the five activities coming from the HHAR data source are used, where walking, walking upstairs and walking downstairs activities are regarded as unknown activities. Table 7.8 shows the average accuracy of discovering unknown and known activities when there is no learning performed.

Activity	State	Average Accuracy (%)
Sit	Known	96
Stand	known	94
Walk	unknown	98
Upstairs	unknown	92
Downstairs	unknown	93

Table 7.8: Average accuracy of discovering unknown and known activities.

   According to results in Table 7.8, the proposed IntElyCare framework recognizes known activities with a mean accuracy of 95%. In addition, most unknown activities are predicted with 94% accuracy on average. Therefore, the results show that IntElyCare can properly discriminate unknown activities from known activities to reinforce activity recognition capability at runtime.

2. **Performance in Learning Unknown Activities on the HHAR Data Source**

   To assess the performance in learning activities, we focus on measuring the accuracy of the proposed IntElyCare framework with and without IConAS that embraces the Continual online activity evolution. The reason to include this experiment is to see whether IConAS presented is effective without interference. To comprehensively validate this effectiveness, 5 cases are conducted, in which an initial activity ontology module that only contains certain activities and an evolved activity ontology module that involves new activities in addition to the initial one, are considered. As for the HHAR data source, the number of activities in the initial activity ontology module is varied from 1 to 4, while the number of activities in the evolved activity ontology module is equal to 5 as shown in Table 7.9. Moreover, in each case, the following

Case	Initial activity ontology module	Evolved activity ontology module
1	sit	sit, stand, walk, upstairs, downstairs
2	sit, stand	sit, stand, walk, upstairs, downstairs
3	sit, stand, walk	sit, stand, walk, upstairs, downstairs
4	sit, stand, walk, upstairs	sit, stand, walk, upstairs, downstairs
5	sit, stand, walk, downstairs	sit, stand, walk, upstairs, downstairs

Table 7.9: A description of the different cases across various known activity settings from the HHAR data source.

activities are performed in the real-world respectively: sitting, standing, walking, walking upstairs, walking and finally walking downstairs.

Table 7.10 compares the accuracy results without and with IConAS under the combination of different types of known activities. From Table 7.10, we observe that the accuracy without Continual online activity evolution suffers from a degradation under influence of performing unknown activities. Also, without the Continual online activity evolution, a possible false prediction of unknown activities can arise, so the overall accuracy can be obviously decreased. In contrast, we can see that the framework, with the Continual online activity evolution offered by the proposed IConAS, tends to improve the accuracy through automatically learning new activities from the HHAR data source at runtime. It is evident from Table 7.10 that IntElyCare achieves high accuracy of 99% after activity learning, which demonstrates the feasibility and effectiveness of IConAS and the Continual online activity evolution when unknown activities are performed at runtime.

	Average accuracy (%)	
Case	**Without IConAS**	**With IConAS**
1	39.89	99
2	53.06	99
3	77.43	99
4	83.63	99
5	87.50	99

Table 7.10: Average accuracy without and with IConAS approach on the HHAR data source.

Apart from accuracy, Figure 7.38 compares the precision obtained without and with IConAS for all cases using the HHAR data source. However, there are significant drops in precision below 90% that are especially evident in the case of the IntElyCare without IConAS. Therefore, our application without IConAS struggles mostly with recognizing unknown activities, which was expected, but IntElyCare with IConAS leads to achieving considerably higher precision in recognizing activities.

Moreover, a similar conclusion can be deduced from the recall results of Figure 7.39. The recall result is considerably better with IConAS, while slightly lower recall without IConAS. Despite this decline in recall result without IConAS, all cases showed significant gains in recall with IConAS.

Furthermore, F1-score results illustrated in Figure 7.40 validate the above findings. Notably, the F1-score with IConAS is higher than 90%. In contrast, a lower F1-score of 90% demonstrates an unsatisfied performance on recognizing activities

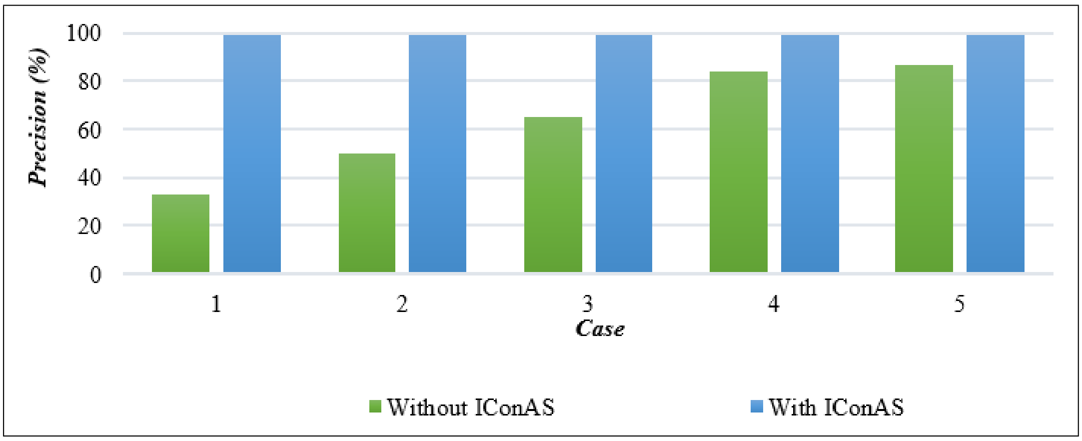

Figure 7.38: Precision scores without and with IConAS approach on the HHAR data source.

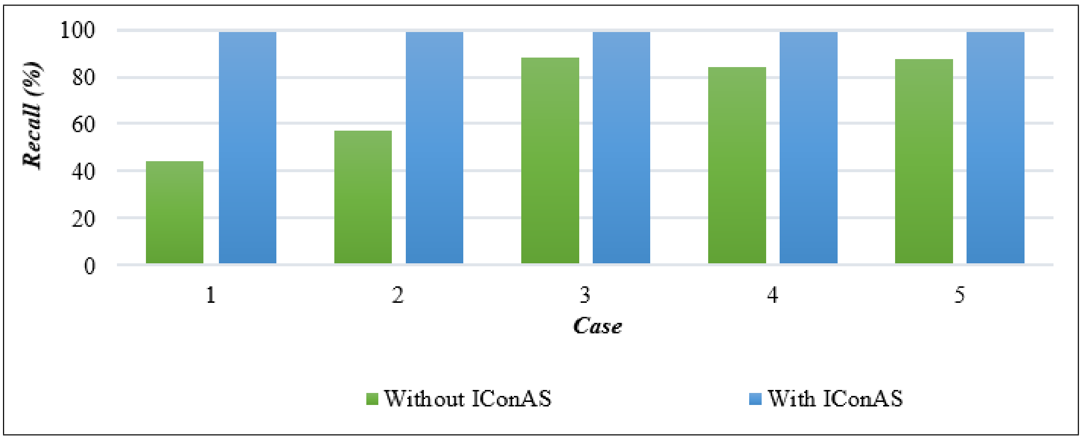

Figure 7.39: Recall scores without and with IConAS approach on the HHAR data source.

without IConAS. Therefore, this means that the IntElyCare framework with IConAS achieves a significant improvement as compared to the IntElyCare without IConAS. By analyzing all results, it is concluded that IntElyCare with IConAS stands out to be the best performance on four metrics; accuracy, precision, recall, and F1-score.

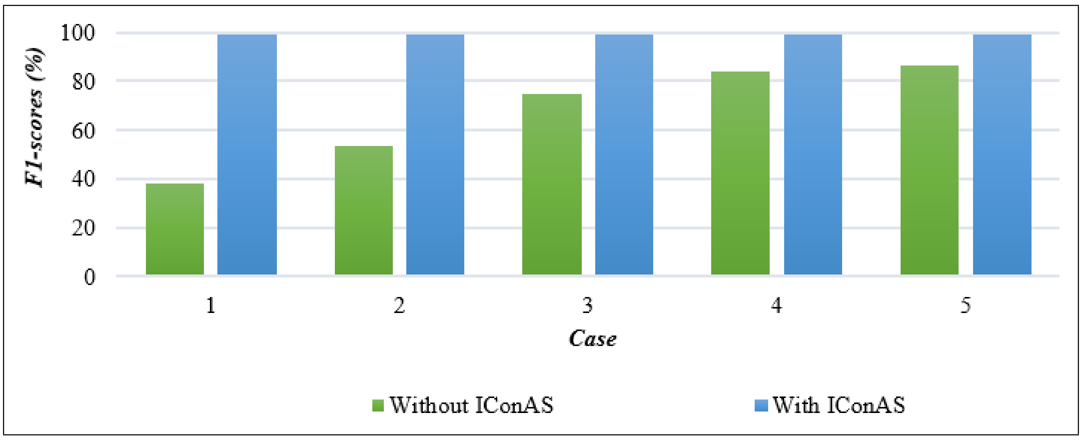

Figure 7.40: F1-scores without and with IConAS approach on the HHAR data source.

7.4.2.3 Activity Recognition Performance Evaluation on the UCI-HAR Data Source

1. **Performance in Learning Unknown Activities on the UCI-HAR Data Source**

   In order to further verify the performance in learning activities, the UCI HAR data source is also employed to measure the accuracy of the IntElyCare framework with and without IConAS. In this experiment, 7 cases are conducted, in which an initial activity ontology module that only contains certain activities and an evolved activity ontology that involves new activities in addition to the initial one, are considered. As for the UCI HAR data source, the number of activities in the initial activity ontology module is varied from 1 to 5, while the number of activities in the evolved activity ontology module is equal to 6 as shown in Table 7.11. Moreover, in each case, the following activities are performed in the real world respectively: sitting, standing, walking, walking upstairs, walking, walking downstairs and finally lying down.

Case	Initial activity ontology module	Evolved activity ontology module
1	sit	sit, stand, walk, upstairs, downstairs, lie
2	sit, stand	sit, stand, walk, upstairs, downstairs, lie
3	sit, stand, walk	sit, stand, walk, upstairs, downstairs, lie
4	sit, stand, walk, upstairs	sit, stand, walk, upstairs, downstairs, lie
5	sit, stand, walk, downstairs	sit, stand, walk, upstairs, downstairs, lie
6	sit, stand, walk, upstairs, lie	sit, stand, walk, upstairs, downstairs
7	sit, stand, walk, downstairs, lie	sit, stand, walk, upstairs, downstairs, lie

Table 7.11: A description of the different cases across various known activity settings from the UCI HAR data source.

Consistent with previous results, the average accuracy with IConAS in Table 7.12 is considerably higher for all cases. We achieved an accuracy of 98.73%, which is an increase in accuracy ranging from almost 7% to 66% compared to the IntElyCare without IConAS. The reason behind such an increase in accuracy is that IntElyCare is able to discriminate among the known and unknown activities and also learn new activities at runtime. The comparison and discussion of obtained results validate that the accuracy of the IntElyCare framework with IConAS is better regardless of the specific data source being used.

	Average accuracy (%)	
Case	**Without IConAS**	**With IConAS**
1	32	98.73
2	48.25	98.73
3	71.04	98.73
4	81.96	98.73
5	86	98.73
6	88.55	98.73
7	91.64	98.73

Table 7.12: Average accuracy without and with IConAS approach on the UCI HAR data source.

Apart from accuracy, Figure 7.41 shows the precision obtained without and with

IConAS for all cases using the UCI HAR data source. The data presented in the figure indicate that the best result is a precision of 98.66%, obtained with IConAS. By comparing the precision results, we see a precision gain with IConAS varied from 8% to 69%.

Moreover, the same conclusion can be reached from the recall results of Figure 7.42. We can clearly see that the IntElyCare framework with IConAS outperform for all cases in terms of recall. Thus, IConAS can improve the recall by 12% - 69%, which provides strong evidence that the proposed Continual online activity evolution is effective across the different known activity settings.

Furthermore, F1-score results that support the prior findings are also presented in Figure 7.43. After analyzing obtained results, we found that the highest F1-score result is gained by using IConAS. For example, in case 5, the result of F1-score without IConAS is 73.2%, while the result with IConAS is improved notably, up to 98.7%, when considering automatic activity evolution at runtime. Overall, we conclude that the IntElyCare with IConAS gain is effective for improving accuracy, precision, recall and F1-score.

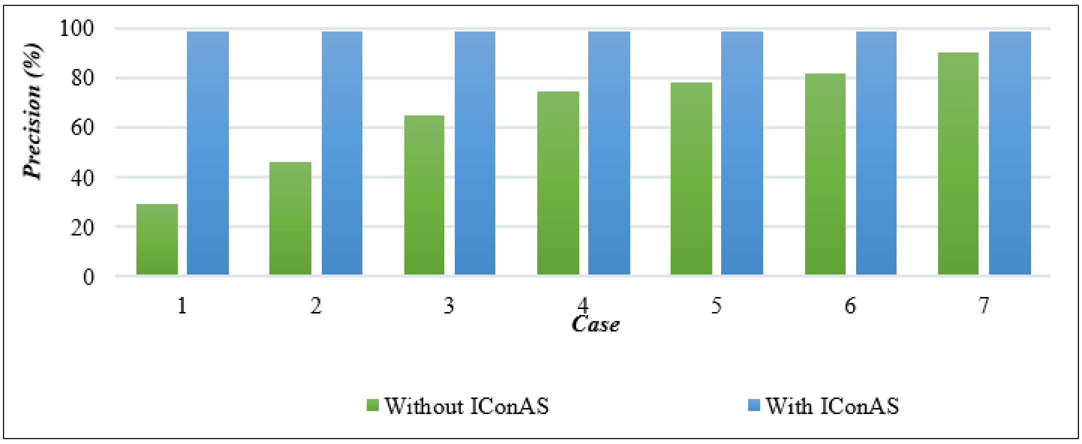

Figure 7.41: Precision scores without and with IConAS approach on the UCI HAR data source.

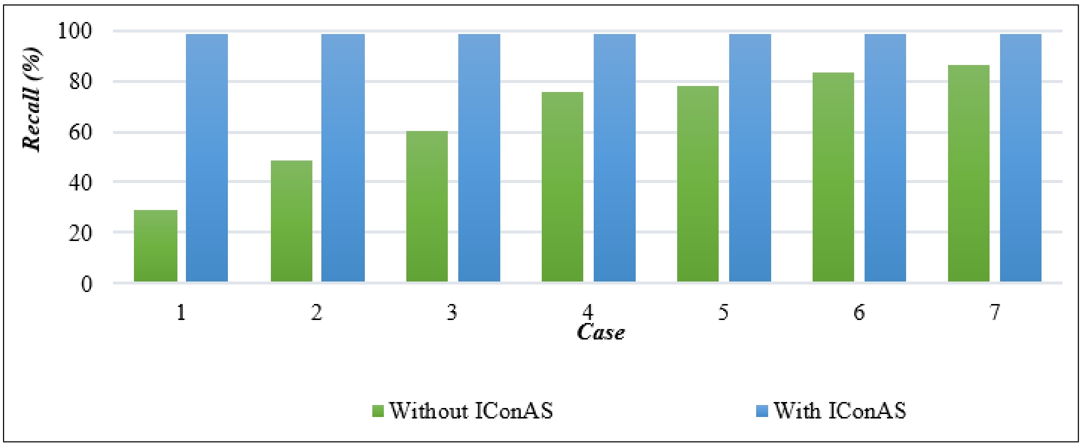

Figure 7.42: Recall scores without and with IConAS approach on the UCI HAR data source.

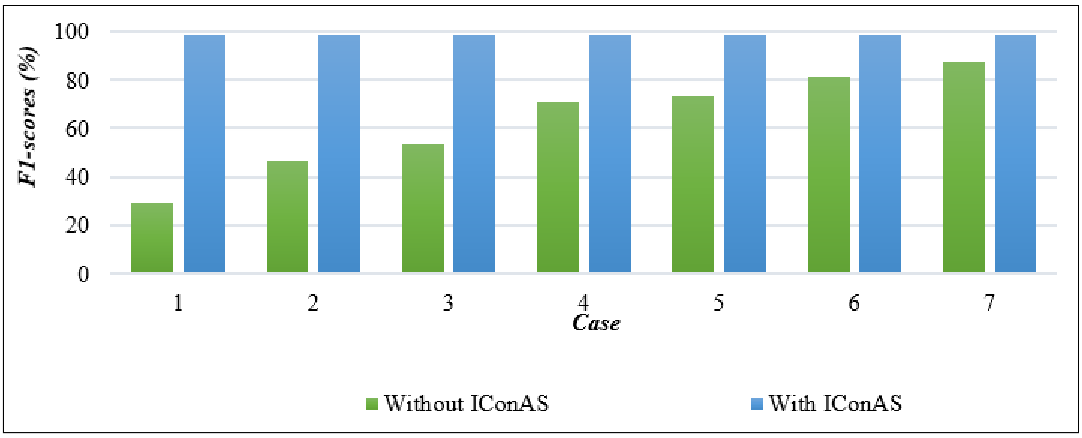

Figure 7.43: F1-scores without and with IConAS approach on the UCI HAR data source.

2. **Performance comparison with existing works using UCI HAR data source**

   In addition to the previous experiments, we also propose an experiment to compare IntElyCare framework accuracy and F1-score with some of existing works from the literature that have used the UCI HAR data source on traditional static setting, in which all activities are assumed known a priori at design time. Works introduced for the purpose of comparison use Machine Learning algorithms, such as, KNN [Jain and Kanhangad, 2017, Mohsen et al., 2021], DT [Cao et al., 2018] and RF [Chetty et al., 2015]. Table 7.13 illustrates the obtained accuracy and F1-score of the proposed IntElyCare framework compared with the existing works [Chetty et al., 2015, Jain and Kanhangad, 2017, Cao et al., 2018, Mohsen et al., 2021] on the UCI HAR data source.

Reference	**Accuracy** (%)	**F1-score** (%)
[Chetty et al., 2015]	96.3	N/A
[Jain and Kanhangad, 2017]	97.12	N/A
[Cao et al., 2018]	94.16	98.6
[Mohsen et al., 2021]	91.46	90.37
IntElyCare	98.73	98.7

Table 7.13: Accuracy and F1-score comparison among some existing works using UCI HAR data source.

Given the results stated in Table 7.13, the proposed IntElyCare obtains higher accuracy of 98.73% compared with the existing works. On the basis of F1-score, it attains a performance of F1-score of 98.7%, which is among the best. Generally, IntElyCare has performed the best across the two performance evaluation criteria.

7.4.3 User Satisfaction Evaluation

User satisfaction is a commonly applied evaluation criterion for services in general. In our case, this criterion assesses the degree of elderly satisfaction with the provided services and decisions. Nevertheless, the evaluation of the elderly satisfaction from their perspectives is a critical step, which can be done with questionnaires. For that, the 8-item Client Satisfaction Questionnaire (CSQ-8) is used [Attkisson and Zwick, 1982] to measure the satisfaction of the elderly for the support they received from the IntElyCare framework. Therefore, the CSQ-8 was used to quantitatively examine elderly participants' experiences with the proposed IntElyCare framework. The CSQ-8 is the short version of the original CSQ-18 [Larsen et al., 1979] and consists of 8 items ranging from Q1 to Q8 used to evaluate elderly satisfaction with provided services, in this case, health notifications. Items were scored on a 4-point Likert scale, with scores ranging from 1 (quite dissatisfied) to 4 (very satisfied), easily scored by summing the participants' item scores to produce a range of 8 to 32, with higher scores indicating greater satisfaction. Total CSQ-8 scores can be subdivided in four categories: poor (8–13); fair (14–19); good (20–25) and excellent (26–32; Smith et al., 2014). An example of the eight different items that the questionnaire measured is "how satisfied are you with the amount of help you have received?" for which the response options are 1 = quite dissatisfied, 2 = indifferent or mildly satisfied, 3 = mostly satisfied, and 4 = very satisfied [Pascoe, 1983]. Reliability of the satisfaction questionnaire was assessed using Cronbach's alpha (α) coefficients, where α of 0.7 or higher were considered acceptable [Nunnally and Bernstein, 1978]. IBM SPSS Statistics, version 28, was used for data analysis.

A total of 100 elderly participants living alone who will use the proposed IntElyCare framework as an aid in their daily life and have heterogeneous context profiles, have accepted to participate in this study. Table 7.14 shows participant characteristics. As illustrated in the table, out of the 100 participants, 53% were males and 72% were in the age range of 60–69. The mean age of the participants was 68 ($\pm$ 5) years. Of the total participants, 64% had chronic disease, 36% were absolutely disease free and 81% were single. After consenting to participate, participants were asked to download and access the IntElyCare for a period of time. Then, they were given a pen and paper questionnaire, which comprised the 8-item as given in Appendix 8.5.

Characteristics	**Numbers**
Gender, n (%)	
• Male	• 53
• Female	• 47
• Age, in years*	• 68 $\pm$ 5
Health status, n (%)	
• With chronic disease	• 64
• Disease free	• 36
Marital status, n (%)	
• Single	• 81
• Married	• 19

Table 7.14: Elderly participants sociodemographic and health characteristics.
* Data presented in mean $\pm$ standard error.

After receiving feedback from the elderly participants, the collected responses were analyzed as outlined in Table 7.15. Regarding the participants' satisfaction, about one-third, 28% of elderly scored in the CSQ-8 "low" satisfaction category (below score 20), whereas 72% were satisfied (total score 20–25) where 22% were very satisfied (total score 26–32). For CSQ-8 items, highest satisfaction rates were efficiency of service received (CSQ-1) 75.3% followed by satisfaction in an overall, general sense, with the service they received

(CSQ-7) 72.9% and recommending friends if they are in need of similar services (CSQ-4) 71%. While lowest rates were obtained with the amount of help, they received (CSQ-5) 70.1% and with the kind of service they needed for personal assistance (CSQ-2) 70.6%. This study showed that the global satisfaction that is measured CSQ-8 was about 71.7% to the delivered services, which is the sum of satisfied and very satisfied, i.e., the total score between 20–25 and 26–32 respectively. We concluded that the majority of elderly was mostly satisfied with IntElyCare. Moreover, descriptive analyses of CSQ-8 showed that the (CSQ-1) item (4.01±0.77), and the (CSQ-7) item (3.39±0.75), had the highest means among all items mean scores. In contrast, the (CSQ-5) item (2.76±0.73) and the (CSQ-2) item (2.99±0.92) had the lowest means among all items mean scores. The average score of (CSQ-8) item was 3.21±0.70, which represents a moderate satisfaction with the offered services. In addition to mean and SD, Cronbach's alpha coefficient was calculated. As shown in Table 7.15, despite differences in some of items mean scores, Cronbach's alpha for all items were satisfactory ranging from 0.70 to 0.90 since α is acceptable at 0.7 and above. Regarding the overall reliability, Cronbach's alpha estimated for the whole questionnaire was equal to 0.82 while it was 0.70 or higher for each item, which underlines the good internal consistency of the questionnaire.

CSQ-8 questionnaire item	Satisfied % (very good/good)	Not satisfied % (poor/fair)	Mean ± SD	Cronbach's alpha
CSQ-1	75.3	24.7	4.01±0.77	0.80
CSQ-2	70.6	29.4	2.99±0.92	0.70
CSQ-3	70.8	29.2	3.02±0.74	0.81
CSQ-4	71	29	3.23±0.44	0.78
CSQ-5	70.1	29.9	2.76±0.73	0.82
CSQ-6	70.8	29.2	3.03±0.74	0.90
CSQ-7	72.9	27.1	3.39±0.75	0.89
CSQ-8	72	28	3.32±0.54	0.90
Total	71.7	28.3	3.21±0.70	0.82

Table 7.15: Satisfaction, mean, standard deviation (SD) and Cronbach's alpha coefficient for CSQ-8.

7.4.4 IConAS Approach Discussion

As noted in the previous experiments, the proposed IntElyCare framework was effective and can outperform existing works. First, IntElyCare reached 94% of average accuracy when dealing with discriminating known and unknown activities. In terms of the unknown-activity discovery, a good accuracy was achieved to demonstrate that the IntElyCare framework is able to truly predict known activities and discover unknown activities. Second, IntElyCare yielded high accuracy, precision, recall and F1-score with rather than without IConAS using both data sources. These results revealed that IntElyCare with IConAS could achieve better performance in learning activities from both data sources, while IntElyCare could struggle without IConAS. While average accuracy gains varied from 9% to 62% with the magnitude of known activities in the initial activity ontology module, we uncovered that IntElyCare is able to reinforce the recognition accuracy by learning new activities and their corresponding decision rules. With high performance for both data sources, it is clear that IntElyCare will be of practical utility to support users in dynamic environments.

Finally, IntElyCare was fairly compared with existing works. Obtained results showed that IntElyCare attains high performance, reaching overall accuracy of 98.73%, and obtains the best F1-score, in comparison with other existing works, which demonstrated IntElyCare is superior in performance. Indicatively, the average accuracy on the UCI HAR data source is 98.73%, 4% higher than the previous works. The results of the comparison allow concluding that IntElyCare is better in terms of performance than the compared works. To this end, the IntElyCare framework could achieve effectiveness in the online activity evolution for real-time activity recognition in dynamic environments, as described in this thesis.

7.5 Concluding Remarks

This chapter has outlined how evaluation studies were carried out and presented the discussion of the achieved results. In particular, evaluation studies presented in this chapter attempted to examine not only the quality of outputs but also the applicability of the design of the outputs. In this sense, first, we have assessed the evolved ontology-based context models that emerged as outcomes of the CoE approach, in different ways, ranging from ontology verification to ontology validation. Second, we have designed to measure the DMA approach effectiveness for each of these parameters Rule Number, Performance, and Computational time. And third, we described the implementation of the proposed IntElyCare framework, where Kafka middleware is augmented in the Elderly Healthcare domain. The IntElyCare framework implementation takes as its starting point the CoE and DMA approaches whose implementations are described in chapter 5 and chapter 6. The implemented IntElyCare framework was evaluated in two rounds of the case-study, where in the second round, the activity recognition performance and the elderly users' satisfaction measuring were considered.

Following up the main achieved results, we found that:

- First, according to the conducted evaluation study described in section 7.2, the CoE approach has shown an appropriate quality of content, schema and structure in the evolved ontology-based context models. It showed a considerable consistency, conciseness and coverage, whereas the completeness was not met that well;
- Second, the findings of the evaluation study, presented in section 7.3, have revealed that the DMA approach could effectively minimize the issues of redundant rule generation, provide a promising performance and a high accuracy to extract a concise set of decision rules even if it takes slightly more time for generating a set of well-performed rules;
- Third, the findings of the evaluation study, provided in section 7.4, have demonstrated the capability of the proposed IntElyCare framework to achieve effectiveness in conducting the Continual online activity evolution in dynamic environments. In addition, according to the result interpretations of the satisfaction study, the IntElyCare framework is able to provide satisfying healthcare decisions and services that could cope with context changes at runtime according to the obtained elderly feedback.

Generally speaking, the evaluations of the different approaches described in this chapter have proven their applicability in different application scenarios in the domain of Elderly

Healthcare and HAR based on results from the multiple evaluations. Despite all that, in our research effort to provide an intelligent context-aware solution that covers runtime changes occurring in dynamic environments, we acknowledge that there is still space for improvement.

CHAPTER 8

Conclusions and Future Works

8.1 Introduction

The final chapter of this thesis concludes this work, presenting the scientific results, limitations and directions for future work. Section 8.2 summarizes the main conclusions of this thesis. Next, an overview of the publications that have emerged from this thesis is illustrated in section 8.3. Then, the limitations and problems faced during the development of this work are presented in section 8.4. Finally, a short discussion of the future directions is given in section 8.5.

8.2 Conclusions

In this thesis, we pointed out problems related to the lack of pervasive middleware and context-aware solutions used to support the runtime changes in users' dynamic environments or in users' preferences and behaviors (chapter 1 and chapter 3). The present thesis introduced a hybrid approach called IConAS to augment existing middleware that led to towards intelligent context-aware solutions. The main purpose of this approach was to consider context evolution and decision-making adaptation in dynamic environments. In this way, this approach combines CoE and DMA approaches for addressing problems that were enumerated in chapter 1. As a result of the development of this approach various tasks can be highlighted. These tasks are the evidence about the achievement of the research objectives as well as the answers of the established research questions. The main tasks are presented in the following:

- Providing an extensive review of the state-of-the-art in pervasive middleware (see chapter 3), giving rise to the identification of some areas for further middleware augmentation (see chapter 4). This was the impetus for the development of the IConAS approach towards an existing middleware augmentation, which presents several advancements to the state-of-the-art with regard to context evolution and decision-making adaptation in a dynamic environment at runtime (see chapter 5 and chapter 6);

- Thoroughly reviewing the literature related to context model evolution approaches and rule learning approaches (see chapter 3);
- Proposing IConAS approach - a hybrid approach which, combines both CoE and DMA approaches for augmenting existing middleware to guarantee the automatic context evolution and decision-making adaptation at runtime (see chapter 4);
- Proposing the IntElyCare framework for the instantiation of IConAS approach in elderly healthcare domain (see chapter 4);
- Proposing the CoE approach capable of automatically evolving an ontology-based context model to reflect runtime context changes in the surrounding dynamic environments (see chapter 5);
- Developing a prototype for the CoE approach (see chapter 5);
- Proposing the DMA approach that allows the enrichment of rule knowledge base through decision rules learning and generation to cope with runtime context changes in the surrounding dynamic environments and context evolution (see chapter 6);
- Developing a prototype for the DMA approach (see chapter 6);
- Performing several experiments with certain criteria, such as, accuracy, precision, recall and F1-scores, tools and participants to assess the proposed approaches. In addition, developing a use case for predefined scenarios, such as the recommendation of healthcare services for elderly (see chapter 7).

In summary, all of these tasks have given rise to augment an existing middleware solution for supporting existing ontology-based context evolution and rule knowledge bases enrichment with new decision rules based on the changes occurring at runtime. Nevertheless, based on the ideas and outcomes discussed in the previous chapters, this work still has some limitations and weaknesses, which can be overcome in future developments.

8.3 Scientific Results

In this section, all publications related to this thesis are listed below. They have been classified according to their type (conferences or journals). Under each publication, we describe the relevance of the forum where it was published.

- **International Journals Indexed in the JCR**
  - Jabla, R., Khemaja, M., Buendia, F., Faiz, S. (2022). Automatic Rule Generation for Decision-Making in Context-Aware Systems Using Machine Learning. Computational Intelligence and Neuroscience, 2022.

    (JCR-SCI: Q1 - Categories: Mathematical Computational, Impact Factor: 3.633).
  - Jabla, R., Khemaja, M., Buendia, F., Faiz, S. (2021). Automatic ontology-based model evolution for learning changes in dynamic environments. Applied Sciences, 11(22), 10770.

(JCR-SCI: Q2 - Categories: Engineering, Multidisciplinary, Impact Factor: 2.679).

- **International Conference and Workshop Papers**
  - Jabla, R., Khemaja, M., Buendia, F., Faiz, S. (2022). A knowledge-driven activity recognition framework for learning unknown activities. Procedia Computer Science, 207, 1871-1880.

    (Core2014 Rank: B).
    * The 26th International Conference on Knowledge-Based and Intelligent Information Engineering Systems (KES 2022);
    * The KES is a Core B conference according to the CORE conference ranking;
    * The conference paper will be published by Elsevier.
  - Jabla, R., Khemaja, M., Buendía, F., Faiz, S. (2022). A Novel Component of Decision-Making for Context-Aware Applications in Pervasive Environments. In: Novais, P., Carneiro, J., Chamoso, P. (eds) Ambient Intelligence – Software and Applications – 12th International Symposium on Ambient Intelligence. ISAmI 2021. Lecture Notes in Networks and Systems, vol 483. Springer, Cham. https://doi.org/10.1007/978-3-031-06894-2_12

    (Scopus indexed).
    * The 12th International Symposium on Ambient Intelligence (ISAmI 2021);
    * The conference paper will be published by Springer (LNCS).
  - Jabla, R., F., Khemaja, Buendía, M. Faiz, S. (2021, September). A Novel Component of Rule Generation in Ubiquitous Computing Environments. In Iberoamerican Workshop on Human-Computer Interaction. CEUR-WS.

    (Scopus indexed).
    * The VII Iberoamerican Conference of Human-Computer Interaction (HCI 2021);
    * The conference paper is published by CEUR-WS.
  - Jabla, R., Khemaja, M., Buend´a, F., Faiz, S. (2020). Smartphone Devices in Smart Environments: Ambient Assisted Living Approach for Elderly People. In the Thirteenth International Conference on Advances in Computer-Human Interactions ACHI, Valencia, Spain (pp. 215-221).

    (Core2014 Rank: C).
    * The Thirteenth International Conference on Advances in Computer-Human Interactions (ACHI 2020);
    * The ACHI is a Core C conference according to the CORE conference ranking.
  - Jabla, R., Buendía, F., Khemaja, M., Faiz, S. (2019). Balancing Timing and Accuracy Requirements in Human Activity Recognition Mobile Applications. In Multidisciplinary Digital Publishing Institute Proceedings (Vol. 31, No. 1, p. 15).

    (Scopus indexed).

    * The 13th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2019);
    * The conference paper is published by Proceedings, which is An Open Access Journal from MDPI.

  - Jabla, R., Braham, A., Buendía, F., Khemaja, M. (2019, June). A Computing Framework to Check Real-Time Requirements in Ambient Intelligent Systems. In International Symposium on Ambient Intelligence (pp. 19-26). Springer, Cham.

    (Scopus indexed).

    * The 10th International Symposium on Ambient Intelligence (ISAmI 2019);
    * The conference paper is published by Springer (LNCS).

8.4 Limitations

While this thesis work has met the goals set forth, obviously, this thesis is subject to certain limitations that should be taken into account. Four dimensions for limitation are identified:

- First of all, our experiment results were limited in terms of the number of participants we invited. Evaluation with a limited number of participants was presented in chapter 7;

- Secondly, we can highlight a limitation in relation to the healthcare services provided. We believe that providing other types of services related to the rest of Maslow's hierarchy levels, such as, the love and belonging level of Maslow's needs, could be enriching the IntElyCare framework with new services;

- Third, the IConAS approach was validated in a practical way only in elderly healthcare domain as presented in chapter 4 and chapter 7. Thus, other domains with different scenarios can be developed to attest the efficiency of this generic characteristic, besides observing the behavior of the proposed IConAS approach in different domains and scenarios. More domains and case studies would benefit current results and improve the applicability of IConAS approach;

- Fourth, a common limitation of the alignment phase in the proposed CoE is that it tends to overlook alignment axioms relating the initial and evolved context models. However, the use of this alignment might or not might introduce inconsistencies in the evolved context models.

8.5 Directions for Future Work

To build up on this research work according to the aforementioned limitations, several potential future lines of research can be considered as follows:

- Inviting more participants from different profiles, in order to have varied group of users (e.g., from different countries, with different level of expertise, etc.);

- Including new services according to the domain in which IConAS approach will be applied. For instance, IConAS approach was validated in elderly healthcare domain, considering the two lowest levels of Maslow's hierarchy. However, other services can be identified with a consideration of the rest of Maslow's hierarchy levels for the improvement of the recommendation process. For this, the enrichment with new healthcare services must be carried out;

- Validating the proposed IConAS approach on more domains, such as smart cities. Furthermore, this validation could be held with a broader range of end-users to obtain further insights into the applicability and generalization of the findings;

- Extending the proposed CoE approach to learn the disjointness and equivalence axioms from external sources during the ontology-based context learning process and then include them in the learned ontology. In particular, for our alignment phase, we propose a new alignment method to determine whether or not the combined axioms are consistent. In this way, the proposed CoE approach will be able to apply the alignment phase through the description logic axioms and to fully guarantee the logical consistency of the evolved context models.

In conclusion, as a result of this research, several novel research opportunities arise that could further strengthen the proposed approaches for more effective intelligent context-aware solutions.

APPENDICES

Appendix 1. Client Satisfaction Questionnaire CSQ-8

Please rate the quality of services you are receiving from IntElyCare on the following:

CSQ-8 items	4: very satisfied	3: quite satisfied	2: neutral	1: quite dissatisfied
CSQ-1. How do you rate the efficiency of services you have received?				
CSQ-2. Did you get the kind of service you expected/needed for personal assistance?				
CSQ-3. To what extent has our application met your health needs and intends?				
CSQ-4. Would you recommend our application to a friend who is in need of similar help?				
CSQ-5. How satisfied are you with the amount of help and assistance you have received through offered services?				
CSQ-6. Have the services you received help you to deal more effectively with your daily situations?				
CSQ-7. In an overall, general sense, how satisfied are you with the services you have received to keep you as independent as possible?				
CSQ-8. Would you reuse our application?				

Thank you for agreeing to complete this questionnaire.

Bibliography

[Abowd, 2012] Abowd, G. D. (2012). What next, ubicomp? celebrating an intellectual disappearing act. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 31–40. See page 22.

[Abowd et al., 1999] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pages 304–307. Springer. See page 13.

[Aggoune, 2018] Aggoune, A. (2018). Automatic ontology learning from heterogeneous relational databases: Application in alimentation risks field. In *IFIP International Conference on Computational Intelligence and Its Applications*, pages 199–210. Springer. See pages 35, 37, and 38.

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216. See page 17.

[Aguado de Cea et al., 2008] Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., and Suárez-Figueroa, M. C. (2008). Natural language-based approach for helping in the reuse of ontology design patterns. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 32–47. Springer. See pages 67, 68.

[Alemdar et al., 2013] Alemdar, H., Ertan, H., Incel, O. D., and Ersoy, C. (2013). Aras human activity datasets in multiple homes with multiple residents. In *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pages 232–235. IEEE. See pages 81, 103, and 108.

[Alm et al., 2009] Alm, N., Astell, A. J., Gowans, G., Dye, R., Ellis, M. P., Vaughan, P., and Riley, P. (2009). Engaging multimedia leisure for people with dementia. *Gerontechnology*. See page 59.

[Almuhareb, 2006] Almuhareb, A. (2006). *Attributes in lexical acquisition*. PhD thesis, University of Essex. See page 67.

[Angelov et al., 2012] Angelov, S., Grefen, P., and Greefhorst, D. (2012). A framework for analysis and design of software reference architectures. *Information and Software Technology*, 54(4):417–431. See page 48.

[Anguita et al., 2013] Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al. (2013). A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, page 3. See page 136.

[Asadianfam et al., 2020] Asadianfam, S., Kolivand, H., and Asadianfam, S. (2020). A new approach for web usage mining using case based reasoning. *SN Applied Sciences*, 2(7):1–11. See pages 40, 43.

[Asim et al., 2018] Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., and Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. *Database*, 2018. See page 35.

[Attkisson and Zwick, 1982] Attkisson, C. C. and Zwick, R. (1982). The client satisfaction questionnaire: Psychometric properties and correlations with service utilization and psychotherapy outcome. *Evaluation and program planning*, 5(3):233–237. See page 143.

[Baldauf et al., 2007] Baldauf, M., Dustdar, S., and Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277. See pages 22, 28.

[Bandyopadhyay et al., 2011] Bandyopadhyay, S., Sengupta, M., Maiti, S., and Dutta, S. (2011). A survey of middleware for internet of things. In *Recent trends in wireless and mobile networks*, pages 288–296. Springer. See page 21.

[Basha, 2021] Basha, M. S. (2021). Early prediction of cardio vascular disease by performing associative classification on medical datasets and using genetic algorithm. In *Intelligent Computing and Innovation on Data Science*, pages 393–402. Springer. See pages 41, 43, 125, 126, and 127.

[Bellavista et al., 2012] Bellavista, P., Corradi, A., Fanelli, M., and Foschini, L. (2012). A survey of context data distribution for mobile ubiquitous systems. *ACM computing surveys (CSUR)*, 44(4):1–45. See pages 14, 15, and 16.

[Bhattacherjee, 2012] Bhattacherjee, A. (2012). Social science research: Principles, methods, and practices. See page 5.

[Boeree, 2006] Boeree, G. C. D. (2006). Personality theories. See page 19.

[Booshehri and Luksch, 2015] Booshehri, M. and Luksch, P. (2015). An ontology enrichment approach by using dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, pages 1–11. See pages 35, 37, and 38.

[Brézillon, 2003] Brézillon, P. (2003). Context dynamic and explanation in contextual graphs. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 94–106. Springer. See page 13.

[Cao et al., 2018] Cao, L., Wang, Y., Zhang, B., Jin, Q., and Vasilakos, A. V. (2018). Gchar: An efficient group-based context—aware human activity recognition on smartphone. *Journal of Parallel and Distributed Computing*, 118:67–80. See page 143.

[Capra et al., 2003] Capra, L., Emmerich, W., and Mascolo, C. (2003). Carisma: Context-aware reflective middleware system for mobile applications. *IEEE Transactions on software engineering*, 29(10):929–945. See pages 23, 33.

[Carroll et al., 2004] Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., and Wilkinson, K. (2004). Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83. See page 90.

[Cederberg and Widdows, 2003] Cederberg, S. and Widdows, D. (2003). Using lsa and noun coordination information to improve the recall and precision of automatic hyponymy extraction. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 111–118. See page 68.

[Celdrán et al., 2014] Celdrán, A. H., Clemente, F. J. G., Pérez, M. G., and Pérez, G. M. (2014). Secoman: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications. *IEEE Systems Journal*, 10(3):1111–1124. See pages 26, 33.

[Chang, 2016] Chang, C. K. (2016). Situation analytics: a foundation for a new software engineering paradigm. *Computer*, 49(1):24–33. See page 90.

[Chansanam et al., 2021] Chansanam, W., Tuamsuk, K., Kwiecien, K., Sutthiprapa, K., and Supnithi, T. (2021). Thai tattoo wisdom's representation of knowledge by ontology. In *Informatics*, volume 8, page 3. Multidisciplinary Digital Publishing Institute. See page 112.

[Chen and Kotz, 2000] Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. See page 13.

[Chen et al., 2004a] Chen, H. et al. (2004a). An intelligent broker architecture for pervasive context-aware systems. See page 23.

[Chen et al., 2003] Chen, H., Finin, T., and Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The knowledge engineering review*, 18(3):197–207. See pages 24, 33.

[Chen et al., 2004b] Chen, H., Perich, F., Finin, T., and Joshi, A. (2004b). Soupa: Standard ontology for ubiquitous and pervasive applications. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, pages 258–267. IEEE. See page 54.

[Chen et al., 2012] Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., and Yu, Z. (2012). Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808. See page 21.

[Chetty et al., 2015] Chetty, G., White, M., and Akther, F. (2015). Smart phone based data mining for human activity recognition. *Procedia Computer Science*, 46:1181–1187. See page 143.

[Cimiano, 2006] Cimiano, P. (2006). *Ontology learning and population from text: algorithms, evaluation and applications*, volume 27. Springer Science & Business Media. See page 16.

[Conti et al., 2012] Conti, M., Das, S. K., Bisdikian, C., Kumar, M., Ni, L. M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G., and Zambonelli, F. (2012). Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence. *Pervasive and mobile computing*, 8(1):2–21. See page 1.

[Cook and Das, 2004] Cook, D. and Das, S. K. (2004). *Smart environments: technology, protocols, and applications*, volume 43. John Wiley & Sons. See page 2.

[Cook and Das, 2007] Cook, D. J. and Das, S. K. (2007). How smart are our environments? an updated look at the state of the art. *Pervasive and mobile computing*, 3(2):53–73. See page 1.

[Da Rocha and Endler, 2012] Da Rocha, R. C. A. and Endler, M. (2012). *Context management for distributed and dynamic context-aware computing*. Springer Science & Business Media. See page 90.

[Davagdorj and Ryu, 2018] Davagdorj, K. and Ryu, K. H. (2018). Association rule mining on head and neck squamous cell carcinoma cancer using fp growth algorithm. In *Proceedings of the International Conference on Information, System and Convergence Applications*. See pages 40, 43.

[de Matos et al., 2017] de Matos, E., Amaral, L. A., Tiburski, R. T., Schenfeld, M. C., de Azevedo, D. F., and Hessel, F. (2017). A sensing-as-a-service context-aware system for internet of things environments. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 724–727. IEEE. See pages 27, 33.

[Dourish and Mainwaring, 2012] Dourish, P. and Mainwaring, S. D. (2012). Ubicomp's colonial impulse. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 133–142. See page 22.

[Duanel, 1977] Duanel, S. (1977). Growth psychology: Models of the healthy personality. *New York: Dvana Nostrand Compan*, 450. See page 19.

[Esseynew et al., 2016] Esseynew, S., Amanuel, B., Sinke, M., Damtew, A., and Koceski, S. (2016). Context-aware technologies and systems. *ON APPLIED INTERNET AND INFORMATION TECHNOLOGIES*, page 120. See pages 14, 15, and 16.

[Evchina et al., 2015] Evchina, Y., Puttonen, J., Dvoryanchikova, A., and Lastra, J. L. M. (2015). Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems. *Engineering Applications of Artificial Intelligence*, 43:111–126. See pages 27, 33.

[Fielding and Taylor, 2002] Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2):115–150. See page 99.

[Flouris et al., 2008] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., and Antoniou, G. (2008). Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(2):117–152. See page 16.

[Freitas, 2000] Freitas, A. A. (2000). Understanding the crucial differences between classification and discovery of association rules: a position paper. *AcM sIGKDD Explorations Newsletter*, 2(1):65–69. See page 41.

[Gabroveanu and Diaconescu, 2008] Gabroveanu, M. and Diaconescu, I.-M. (2008). Extracting semantic annotations from moodle data. *Programme Chairs*, 1. See pages 40, 41, and 43.

[Gad et al., 2018] Gad, I., Manjunatha, B., et al. (2018). Multi-label classification of big ncdc weather data using deep learning model. In *International Conference on Soft Computing Systems*, pages 232–241. Springer. See page 137.

[Gangemi et al., 2005] Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. (2005). A theoretical framework for ontology evaluation and validation. In *SWAP*, volume 166, page 16. See page 111.

[Garg, 2013] Garg, N. (2013). *Apache kafka*. Packt Publishing Birmingham, UK. See page 51.

[Goldberg, 1989] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. addison. *Reading*. See page 90.

[Gómez-Pérez, 2004] Gómez-Pérez, A. (2004). Ontology evaluation. In *Handbook on ontologies*, pages 251–273. Springer. See page 111.

[Gómez-Pérez et al., 2006] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2006). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media. See page 16.

[Gómez-Pérez et al., 2003] Gómez-Pérez, A., Manzano-Macho, D., et al. (2003). A survey of ontology learning methods and techniques. *OntoWeb Deliverable D*, 1(5). See page 35.

[Grefenstette, 1993] Grefenstette, J. J. (1993). Genetic algorithms and machine learning. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 3–4. See page 94.

[Gu et al., 2004] Gu, T., Pung, H. K., and Zhang, D. Q. (2004). A middleware for building context-aware mobile services. In *2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No. 04CH37514)*, volume 5, pages 2656–2660. IEEE. See pages 24, 33.

[Guarino and Welty, 2002] Guarino, N. and Welty, C. (2002). Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65. See page 108.

[Haase and Stojanovic, 2005] Haase, P. and Stojanovic, L. (2005). Consistent evolution of owl ontologies. In *European Semantic Web Conference*, pages 182–197. Springer. See pages 16, 35.

[Haldulakar and Agrawal, 2011] Haldulakar, R. and Agrawal, J. (2011). Optimization of association rule mining through genetic algorithm. *International Journal on Computer Science and Engineering (IJCSE)*, 3(3):1252–1259. See page 94.

[Haller et al., 2019] Haller, A., Janowicz, K., Cox, S. J., Lefrançois, M., Taylor, K., Le Phuoc, D., Lieberman, J., García-Castro, R., Atkinson, R., and Stadler, C. (2019). The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1):9–32. See page 54.

[Hearst, 1998] Hearst, M. (1998). Wordnet: An electronic lexical database and some of its applications. *Automated Discovery of WordNet Relations*. See page 68.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*. See page 68.

[Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, pages 75–105. See page 5.

[Hoareau and Satoh, 2009] Hoareau, C. and Satoh, I. (2009). Modeling and processing information for context-aware computing: A survey. *New Generation Computing*, 27(3):177–196. See page 2.

[Hodaie et al., 2013] Hodaie, Z., Mannov, N., Zamarripa, S., Van Der Sype, Y. S., Burvall, M., and Maalej, W. (2013). D6. 1-conceptual model of user observation, application monitoring tools, and context actuators. See pages 14, 15.

[Hong et al., 2009] Hong, J., Suh, E.-H., Kim, J., and Kim, S. (2009). Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, 36(4):7448–7457. See pages 40, 41, 43, 125, 126, 127, and 129.

[Hoque et al., 2017] Hoque, M. R., Kabir, M. H., and Yang, S.-H. (2017). Development of a cooperative middleware to provide context-aware service in smart home. *International Journal of Smart Home*, 11(5):33–40. See pages 27, 33.

[Hussain and Abidi, 2007] Hussain, S. and Abidi, S. S. R. (2007). Ontology driven cpg authoring and execution via a semantic web framework. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 135–135. IEEE. See page 97.

[Jabla et al., 2020] Jabla, R., Buendía, F., Khemaja, M., and Faiz, S. (2020). Smartphone devices in smart environments: Ambient assisted living approach for elderly people. See pages 54, 58.

[Jabla et al., 2021a] Jabla, R., Khemaja, M., Buendia, F., and Faiz, S. (2021a). Automatic ontology-based model evolution for learning changes in dynamic environments. *Applied Sciences*, 11(22):10770. See pages 46, 62.

[Jabla et al., 2021b] Jabla, R., Khemaja, M., Buendía, F., and Faiz, S. (2021b). A novel component of rule generation in ubiquitous computing environments. See pages 46, 90.

[Jabla et al., 2022a] Jabla, R., Khemaja, M., Buendia, F., and Faiz, S. (2022a). Automatic rule generation for decision-making in context-aware systems using machine learning. *Computational Intelligence and Neuroscience*, 2022. See pages 46, 90.

[Jabla et al., 2022b] Jabla, R., Khemaja, M., Buendia, F., and Faiz, S. (2022b). A knowledge-driven activity recognition framework for learning unknown activities. *Procedia Computer Science*, 207:1871–1880. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES2022. See page 49.

[Jabla et al., 2022c] Jabla, R., Khemaja, M., Buendía, F., and Faiz, S. (2022c). A novel component of decision-making for context-aware applications in pervasive environments. In Novais, P., Carneiro, J., and Chamoso, P., editors, *Ambient Intelligence – Software and Applications – 12th International Symposium on Ambient Intelligence*, pages 127–137, Cham. Springer International Publishing. See page 46.

[Jain and Kanhangad, 2017] Jain, A. and Kanhangad, V. (2017). Human activity classification in smartphones using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 18(3):1169–1177. See page 143.

[Kaliappan et al., 2019] Kaliappan, J., Sai, S., et al. (2019). Weblog and retail industries analysis using a robust modified apriori algorithm. See pages 40, 43.

[Khadir et al., 2021] Khadir, A. C., Aliane, H., and Guessoum, A. (2021). Ontology learning: Grand tour and challenges. *Computer Science Review*, 39:100339. See page 36.

[Klancnik and Blazic, 2010] Klancnik, T. and Blazic, B. (2010). Context-aware information broker for cloud computing. *International Review on Computer and Software*, 5(1). See page 35.

[Kotsiantis et al., 2006] Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al. (2006). Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, 30(1):25–36. See page 17.

[Kuntarto et al., 2019] Kuntarto, G. P., Alrin, Y., and Gunawan, I. P. (2019). The key role of ontology alignment and enrichment methodologies for aligning and enriching dwipa ontology with the weather concept on the tourism domain. In *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–6. IEEE. See pages 35, 37, and 38.

[Lantow, 2016] Lantow, B. (2016). Ontometrics: Putting metrics into use for ontology evaluation. In *KEOD*, pages 186–191. See page 108.

[Lara and Labrador, 2012] Lara, O. D. and Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209. See page 49.

[Larsen et al., 1979] Larsen, D. L., Attkisson, C. C., Hargreaves, W. A., and Nguyen, T. D. (1979). Assessment of client/patient satisfaction: development of a general scale. *Evaluation and program planning*, 2(3):197–207. See page 144.

[Lehmann and Voelker, 2014] Lehmann, J. and Voelker, J. (2014). An introduction to ontology learning. *Perspectives on Ontology Learning*, pages 7–14. See page 35.

[Li et al., 2015] Li, X., Eckert, M., Martinez, J.-F., and Rubio, G. (2015). Context aware middleware architectures: Survey and challenges. *Sensors*, 15(8):20570–20607. See page 33.

[Liu, 2011] Liu, B. (2011). *Web data mining: exploring hyperlinks, contents, and usage data*, volume 1. Springer. See page 18.

[Liu et al., 2014] Liu, Q., Logan, J., Tian, Y., Abbasi, H., Podhorszki, N., Choi, J. Y., Klasky, S., Tchoua, R., Lofstead, J., Oldfield, R., et al. (2014). Hello adios: the challenges and lessons of developing leadership class i/o frameworks. *Concurrency and Computation: Practice and Experience*, 26(7):1453–1473. See page 62.

[Lucene, 2010] Lucene, A. (2010). Apache lucene-overview. *Internet: http://lucene. apache. org/iava/docs/[Jan. 15, 2009]*. See page 74.

[Ma and Molnár, 2020] Ma, C. and Molnár, B. (2020). Use of ontology learning in information system integration: A literature survey. In *ACIIDS (Companion)*, pages 342–353. See page 35.

[Madhusudanan et al., 2018] Madhusudanan, J., Geetha, S., Prasanna Venkatesan, V., Vignesh, U., and Iyappan, P. (2018). Hybrid aspect of context-aware middleware for pervasive smart environment: A review. *Mobile Information Systems*, 2018. See pages 22, 28.

[Mahmood et al., 2013] Mahmood, A., Shi, K., Khatoon, S., and Xiao, M. (2013). Data mining techniques for wireless sensor networks: A survey. *International Journal of Distributed Sensor Networks*, 9(7):406316. See page 41.

[March and Smith, 1995] March, S. T. and Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, 15(4):251–266. See page 5.

[Martin et al., 2004] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., et al. (2004). Owl-s: Semantic markup for web services. *W3C member submission*, 22(4). See page 55.

[Maslow, 1981] Maslow, A. H. (1981). *Motivation and personality*. Prabhat Prakashan. See page 19.

[Maslow, 1998] Maslow, A. H. (1998). A theory of human motivation. *Personality: Critical concepts in psychology*, pages 169–188. See page 18.

[McBride, 2002] McBride, B. (2002). Jena: A semantic web toolkit. *IEEE Internet computing*, 6(6):55–59. See page 16.

[Meditskos et al., 2016] Meditskos, G., Dasiopoulou, S., and Kompatsiaris, I. (2016). Metaq: A knowledge-driven framework for context-aware activity recognition combining sparql and owl 2 activity patterns. *Pervasive and Mobile Computing*, 25:104–124. See page 54.

[Mikalsen et al., 2006] Mikalsen, M., Paspallis, N., Floch, J., Stav, E., Papadopoulos, G. A., and Chimaris, A. (2006). Distributed context management in a mobility and adaptation enabling middleware (madam). In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 733–734. See page 25.

[Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41. See page 71.

[Miswan et al., 2021] Miswan, N. H., Sulaiman, [U+FFFD]M., Chan, C. S., and Ng, C. G. (2021). Association rules mining for hospital readmission: A case study. *Mathematics*, 9(21):2706. See pages 40, 41, and 43.

[Mohsen et al., 2021] Mohsen, S., Elkaseer, A., and Scholz, S. G. (2021). Human activity recognition using k-nearest neighbor machine learning algorithm. In *Proceedings of the International Conference on Sustainable Design and Manufacturing*, pages 304–313. Springer. See page 143.

[Moore et al., 2007] Moore, P., Hu, B., Zhu, X., Campbell, W., and Ratcliffe, M. (2007). A survey of context modeling for pervasive cooperative learning. In *2007 First IEEE International Symposium on Information Technologies and Applications in Education*, pages K5–1. IEEE. See pages 14, 15, and 16.

[Morgner et al., 2017] Morgner, P., Müller, C., Ring, M., Eskofier, B., Riess, C., Armknecht, F., and Benenson, Z. (2017). Privacy implications of room climate data. In *European Symposium on Research in Computer Security*, pages 324–343. Springer. See page 108.

[Musen et al., 2006] Musen, M., Crubézy, M., Fergerson, R., Noy, N., Tu, S., and Vendetti, J. (2006). The protégé ontology editor and knowledge acquisition system. *Stanford Center for Biomedical Informatics Research*. See page 16.

[Newman, 2021] Newman, S. (2021). *Building microservices*. " O'Reilly Media, Inc.". See page 132.

[Nunnally and Bernstein, 1978] Nunnally, J. C. and Bernstein, I. (1978). Psychometric theory mcgraw-hill new york. *The role of university in the development of entrepreneurial vocations: a Spanish study*, pages 387–405. See page 144.

[Ogunde et al., 2011] Ogunde, A., Folorunso, O., Sodiya, A., Oguntuase, J., and Ogunleye, G. (2011). Improved cost models for agent-based association rule mining in distributed databases. *Anale. SeriaInformatica*, 9(1):231–250. See page 17.

[Ordónez et al., 2013] Ordónez, F. J., De Toledo, P., and Sanchis, A. (2013). Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors*, 13(5):5460–5477. See pages 78, 100, and 108.

[Parekh et al., 2015] Parekh, A., Patel, A., Parmar, S., and Patel, V. (2015). Web usage mining: frequent pattern generation using association rule mining and clustering. *Int. J. Eng. Res. Technol*, 4(4):1243–1246. See page 18.

[Pascoe, 1983] Pascoe, G. C. (1983). Patient satisfaction in primary health care: a literature review and analysis. *Evaluation and program planning*, 6(3-4):185–210. See page 144.

[Perera et al., 2013] Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2013). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454. See pages 22, 28.

[Petasis et al., 2011] Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., and Zavitsanos, E. (2011). Ontology population and enrichment: State of the art. *Knowledge-driven multimedia information extraction and ontology evolution*, pages 134–166. See page 16.

[Poslad, 2011] Poslad, S. (2011). *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons. See page 12.

[Pottinger and Bernstein, 2003] Pottinger, R. A. and Bernstein, P. A. (2003). Merging models based on given correspondences. In *Proceedings 2003 VLDB Conference*, pages 862–873. Elsevier. See page 73.

[Poveda et al., 2009] Poveda, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2009). Common pitfalls in ontology development. In *Conference of the Spanish Association for Artificial Intelligence*, pages 91–100. Springer. See page 110.

[Poveda Villalón, 2016] Poveda Villalón, M. (2016). *Ontology Evaluation: a pitfall-based approach to ontology diagnosis*. PhD thesis, ETSI_Informatica. See pages 111, 112.

[Poveda-Villalón et al., 2014] Poveda-Villalón, M., Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2014). Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34. See page 108.

[Poveda Villalon et al., 2010] Poveda Villalon, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2010). A double classification of common pitfalls in ontologies. See page 110.

[Poveda-Villalón et al., 2012] Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2012). Validating ontologies with oops! In *International conference on knowledge engineering and knowledge management*, pages 267–281. Springer. See page 73.

[Pradeep and Krishnamoorthy, 2019] Pradeep, P. and Krishnamoorthy, S. (2019). The mom of context-aware systems: A survey. *Computer Communications*, 137:44–69. See pages 22, 28.

[Pradeep et al., 2021] Pradeep, P., Krishnamoorthy, S., and Vasilakos, A. V. (2021). A holistic approach to a context-aware iot ecosystem with adaptive ubiquitous middleware. *Pervasive and Mobile Computing*, 72:101342. See pages 27, 33.

[Qodmanan et al., 2011] Qodmanan, H. R., Nasiri, M., and Minaei-Bidgoli, B. (2011). Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Systems with applications*, 38(1):288–298. See page 95.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106. See page 92.

[Román et al., 2002] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., and Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE pervasive computing*, 1(4):74–83. See pages 22, 33.

[Rouvoy et al., 2008] Rouvoy, R., Beauvois, M., Lozano, L., Lorenzo, J., and Eliassen, F. (2008). Music: an autonomous platform supporting self-adaptive mobile applications. In *Proceedings of the 1st workshop on Mobile middleware: embracing the personal communication device*, pages 1–6. See pages 25, 33.

[Rubini and Eswaran, 2015] Rubini, L. J. and Eswaran, P. (2015). Generating comparative analysis of early stage prediction of chronic kidney disease. *International Journal of Modern Engineering Research (IJMER)*, 5(7):49–55. See page 124.

[Saad and Nagarur, 2020] Saad, H. and Nagarur, N. (2020). Data mining techniques in predicting breast cancer. *arXiv preprint arXiv:2011.11088*. See page 92.

[Salber et al., 1999] Salber, D., Dey, A. K., and Abowd, G. D. (1999). The context toolkit: Aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 434–441. See pages 22, 33.

[Sarath and Ravi, 2013] Sarath, K. and Ravi, V. (2013). Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 26(8):1832–1840. See page 94.

[Sarker, 2019] Sarker, I. H. (2019). A machine learning based robust prediction model for real-life mobile phone data. *Internet of Things*, 5:180–193. See pages 41, 43, 125, 126, 127, and 129.

[Sbissi et al., 2020] Sbissi, S., Mahfoudh, M., and Gattoufi, S. (2020). A medical decision support system for cardiovacsular disease based on ontology learning. In *2020 International Multi-Conference on:"Organization of Knowledge and Advanced Technologies"(OCTA)*, pages 1–9. IEEE. See pages 35, 37, and 38.

[Schultz and Schultz, 2016] Schultz, D. P. and Schultz, S. E. (2016). *Theories of personality*. Cengage Learning. See page 19.

[Shamsfard and Barforoush, 2003] Shamsfard, M. and Barforoush, A. A. (2003). The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(4):293–316. See page 36.

[Simon, 1996] Simon, H. A. (1996). *The sciences of the artificial*. MIT press. See page 5.

[Small et al., 1999] Small, S. A., Stern, Y., Tang, M., and Mayeux, R. (1999). Selective decline in memory function among healthy elderly. *Neurology*, 52(7):1392–1392. See page 59.

[Sowa, 1999] Sowa, J. F. (1999). *Knowledge representation: logical, philosophical and computational foundations*. Brooks/Cole Publishing Co. See page 67.

[Stisen et al., 2015] Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. (2015). Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 127–140. See pages 76, 99, 108, and 133.

[Stoilos et al., 2005] Stoilos, G., Stamou, G., and Kollias, S. (2005). A string metric for ontology alignment. In *International Semantic Web Conference*, pages 624–637. Springer. See page 71.

[Suárez-Figueroa, 2010] Suárez-Figueroa, M. C. (2010). *NeOn Methodology for building ontology networks: specification, scheduling and reuse*. PhD thesis, Informatica. See page 107.

[Suárez-Figueroa et al., 2012] Suárez-Figueroa, M. C., Gómez-Pérez, A., and Fernández-López, M. (2012). The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer. See page 16.

[Suresh et al., 2020] Suresh, A., Udendhran, R., and Balamurgan, M. (2020). Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers. *Soft Computing*, 24(11):7947–7953. See page 93.

[Surve and Ghorpade, 2017] Surve, A. R. and Ghorpade, V. R. (2017). Pervasive context-aware computing survey of context-aware ubiquitous middleware systems. *International Journal of Engineering Research and Technology*, 10(1):974–3154. See pages 14, 15, and 16.

[Tamang, 2015] Tamang, T. M. (2015). User-centered design of an interactive social service concept for elderly people. Master's thesis. See pages 18, 58.

[Tartir et al., 2005] Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., and Aleman-Meza, B. (2005). Ontoqa: Metric-based ontology quality analysis. See page 108.

[Thielke et al., 2012] Thielke, S., Harniss, M., Thompson, H., Patel, S., Demiris, G., and Johnson, K. (2012). Maslow's hierarchy of human needs and the adoption of health-related technologies for older adults. *Ageing international*, 37(4):470–488. See page 18.

[Tzanavari et al., 2008] Tzanavari, A., Sielis, G., Dolog, P., Kouloumbis, A., and Schmid, K. (2008). Deliverable d3. 1-description of context awareness in idspace: idspace project deliverable. See pages 14, 16.

[Vaizman et al., 2017] Vaizman, Y., Ellis, K., and Lanckriet, G. (2017). Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE pervasive computing*, 16(4):62–74. See page 108.

[Villegas and Müller, 2010] Villegas, N. M. and Müller, H. A. (2010). Managing dynamic context to optimize smart interactions and services. *The smart internet*, pages 289–318. See pages 14, 16.

[Walls, 2015] Walls, C. (2015). *Spring Boot in action*. Simon and Schuster. See page 131.

[Wei and Chan, 2013] Wei, E. J. and Chan, A. T. (2013). Campus: A middleware for automated context-aware adaptation decision making at run time. *Pervasive and Mobile Computing*, 9(1):35–56. See pages 26, 33.

[Wei and Jin, 2012] Wei, Q. and Jin, Z. (2012). Service discovery for internet of things: a context-awareness perspective. In *Proceedings of the fourth Asia-Pacific symposium on internetware*, pages 1–6. See pages 14, 15, and 16.

[Weiser, 1991] Weiser, M. (1991). The computer for the 21 st century. *Scientific american*, 265(3):94–105. See pages 1, 11, and 12.

[Weiser and Brown, 1997] Weiser, M. and Brown, J. S. (1997). The coming age of calm technology. In *Beyond calculation*, pages 75–85. Springer. See page 1.

[Weiser et al., 1999] Weiser, M., Gold, R., and Brown, J. S. (1999). The origins of ubiquitous computing research at parc in the late 1980s. *IBM systems journal*, 38(4):693–696. See pages 1, 11.

[Weyrich, 1999] Weyrich, C. (1999). Information society technologies advisory group orientations for workprogramme 2000 and beyond. See page 12.

[Wieringa, 2009] Wieringa, R. (2009). Design science as nested problem solving. In *Proceedings of the 4th international conference on design science research in information systems and technology*, pages 1–12. See pages 5, 6, and 21.

[Wieringa, 2014] Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer. See pages 5, 6, and 21.

[Witten and Frank, 2002] Witten, I. H. and Frank, E. (2002). Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, 31(1):76–77. See page 92.

[Witten et al., 1999] Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., and Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with java implementations. See page 123.

[Yao et al., 2014] Yao, Y., Liu, H., Yi, J., Chen, H., Zhao, X., and Ma, X. (2014). An automatic semantic extraction method for web data interchange. In *2014 6th International Conference on Computer Science and Information Technology (CSIT)*, pages 148–152. IEEE. See pages 35, 37, and 38.

[Zulkernain et al., 2010] Zulkernain, S., Madiraju, P., Ahamed, S. I., and Stamm, K. (2010). A mobile intelligent interruption management system. *Journal of Universal Comupter Science*. See pages 40, 41, and 43.