Semantically Steered Clinical Decision Support Systems

By

Eider Sanchez

Submitted to the department of Computer Science and Artificial Intelligence in partial fulfillment of the requirements for the degree of Doctor of Philosophy

At
The University of the Basque Country
Donostia - San Sebastian
2014

Universidad del País Vasco / Euskal Herriko Unibertsitatea

AUTORIZACION DEL/LA DIRECTOR/A DE TESIS

PARA SU PRESENTACION

Dr/a. _______________________________________ con N.I.F. ________________________

como Director/a de la Tesis Doctoral: ____

____

____

realizada en el Departamento ____

____

por el Doctorando Don/ña. ____ ,

autorizo la presentación de la citada Tesis Doctoral, dado que reúne las condiciones necesarias para su defensa.

En ________________ a _____ de __________________ de _____

EL/LA DIRECTOR/A DE LA TESIS

Fdo.: ______________________________

AUTORIZACION DEL/LA DIRECTOR/A DE TESIS

PARA SU PRESENTACION

Dr/a. ____con N.I.F.________________________

como Director/a de la Tesis Doctoral: ____

____

____

realizada en el Departamento ____

____

por el Doctorando Don/ña. ____ ,

autorizo la presentación de la citada Tesis Doctoral, dado que reúne las condiciones necesarias para su defensa.

En _________________a _____de ___________________de _____

EL/LA DIRECTOR/A DE LA TESIS

Fdo.: ________________________________

Universidad del País Vasco — Euskal Herriko Unibertsitatea

CONFORMIDAD DEL DEPARTAMENTO

El Consejo del Departamento de ____

____

en reunión celebrada el día ____ de ____________ de _____ ha acordado dar la conformidad a la admisión a trámite de presentación de la Tesis Doctoral titulada: ___________

____

____

dirigida por el/la Dr/a. ____

____

y presentada por Don/ña. ____

ante este Departamento.

En _________________ a _____de ____________________de _____

Vº Bº DIRECTOR/A DEL DEPARTAMENTO

SECRETARIO/A DEL DEPARTAMENTO

Fdo.: __________________________________

Fdo.: ______________________________

emun ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

ACTA DE GRADO DE DOCTOR

ACTA DE DEFENSA DE TESIS DOCTORAL

DOCTORANDO DON/ÑA. ____

TITULO DE LA TESIS: ____

____

____

El Tribunal designado por la Subcomisión de Doctorado de la UPV/EHU para calificar la Tesis Doctoral arriba indicada y reunido en el día de la fecha, una vez efectuada la defensa por el doctorando y contestadas las objeciones y/o sugerencias que se le han formulado, ha otorgado por___________________la calificación de:
unanimidad ó mayoría

Idioma/s defensa: ____

En _________________a _____de ____________________de _____

EL/LA PRESIDENTE/A, EL/LA SECRETARIO/A,

Fdo.: Fdo.:

Dr/a: ____________________ Dr/a: ______________________

VOCAL 1º, VOCAL 2º, VOCAL 3º,

Fdo.: Fdo.: Fdo.:

Dr/a: ____________________Dr/a:_______________________Dr/a:____________________

EL/LA DOCTORANDO/A,

Fdo.: ______________________

Semantically Steered Clinical Decision Support Systems

by

Eider Sanchez

Submitted to the Department of Computer Science and Artificial Intelligence on February 18, 2014,
in partial fulfillment of the requirements for the degree of Doctor of Philosophy

abstract

Clinical Decision Support Systems (CDSS) are currently a hot topic of research, offering the possibility of enhanced health care services and optimized management of resources. This doctoral dissertation provides an innovative architecture for semantically steered CDSS. We propose the use of domain modeling paradigms for enhancing classical CDSS with a Knowledge Engineering approach, coining the term S-CDSS. Our work focuses on the practical aspects of decision-making, commonly present in daily clinical practice. This Thesis contains contributions, such as (i) the use of pre-cached Knowledge, (ii) the generation of new architectures for clinical decision-making, and (iii) the Knowledge persistence during the clinical life cycle. Fundamental to our work is the pioneering use of the modeling and re-use of physicians' experience that leads towards a repository of the decisions performed. We have implemented our approach in two application domains within industrial projects developed in real world clinical environments: the early diagnosis of Alzheimer's Disease, and the diagnosis, treatment and follow-up of Breast Cancer.

To Aitor

Originality Statement

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at The University of the Basque Country or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at The University of the Basque Country or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.
The author hereby grants to University of the Basque Country permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signed:____

"Be sceptical, ask questions, demand proof. Demand evidence. Don't take anything for granted. But here's the thing: When you get proof, you need to accept the proof. And we're not that good at doing that."

Michael Specter, TED Talks, February 2010

Acknowledgements

I would like to thank to Prof. Manuel Grana and Dr. Carlos Toro for your support during these years and for the opportunity you gave me to start my research career. It has been a real pleasure to learn from you. I would also like to thank to Dr. Celine Paloc, Dr. Shabs Rajasekharan and Dr. Ivan Macia for your trust on me.Thank you also to Eduardo Carrasco, for everything you have taught me these years, and to Arkaitz Artetxe for your collaboration, specially during the MIND project. Jon Haitz Legarreta, Ane Murua, Gorka Epelde, Dr. Inigo Barandiaran, Dr. Luis Kabongo, Dr. Andoni Beristain, Dr. Greg Maclair, Dr. Alessandro de Mauro and the rest of the Biomedical Applications department, thank you for your collaboration. I would also like to thank to Dr. Agustin Agirre your trust on my work. Dr. Amparo Gonzalez, Dr. Luis Brualla and Dr. Jose Carlos Gordo, thank you for the interesting work you have shared with me. I would also like to thank to the girls (Ainhoa, Esther, Maider and Marta) for making me laugh each and every day. Vicomtech would not be the same without you. And, of course, to my family, thank you for being there every day when I needed you. Esti, Amaia I., Olatz, Amaia A., Ilazki, Maddalen I., Maddalen A., Itziar, Zureya and Ane, you are like a second family for me. Finally, an especial thank you for Aitor, for showing me the best of life.

Contents

List of Figures

List of Tables

Part I

Introduction and foundations

Chapter 1

Introduction and Overview

This chapter provides a general introduction to this Thesis, presenting a quick overview to its contents, objectives, contributions, related publications and structure.

Section 1.1 presents our motivations; Section 1.2 enumerates the objectives of this Thesis; Section 1.3 sumarizes the methodological and technical contributions; Section 1.4 describes the research environment and context where this Thesis was developed, as well as the publications derived. Lastly, section 1.5 details the chapters structure.

1.1 Thesis motivation

The human factor is decisive in the success of clinical decisions. It is of the outmost importance when reasoning capabilities and prior knowledge on the problem are considered at the moment when those decisions are made [SB]. During the last years several studies have discussed human errors in medicine relating its overall impact to the health system [CN11, dVRS+08, KCD00, Joh07, BLL+91]. According to [KCD00] it was estimated that between 44.000 and 98.000 patients died every year in the 1990's in US due to medical errors. The derived costs exceeded the 17 and 29 billion American dollars respectively. Brennan et al. [BLL+91] argued that about a 50% of those errors were preventable, making the aforesaid statistics even more impacting.

At the beginning of the 2000's, the patient safety movement became stronger, due in part to the influence of other sectors such as aviation or nuclear power, were the tolerated failure rates were in comparison extremely lower [Joh07]. Meanwhile, human error in the health system was believed at that time as inevitable. Given that scenario, during the last years, great efforts have been made for the development and the implementation of solutions aimed towards the reduction of the incidence and impact of preventable medical errors [Joh07].

In particular, tools and computer systems for supporting decision making are some of the most relevant efforts made. So called Clinical Decision Support Systems (CDSS), are active knowledge resources that use patient clinical data to generate case specific advice [LWA06]. CDSS (i) gather and analyze patient clinical and family history, patient data coming from medical devices, evidence provided by the medical community, as well as hospital specific accepted guidelines for each case, and (ii) provide recommendations to physicians in order that they have the necessary knowledge to make a proper decision.

Numerous CDSS and technologies have been proposed [Blo12, Hol08, WS08, BL07], but the integration of such systems in daily clinical environments has not been fully achieved yet [OTM+07]. Some authors [PT06, SWO+08, KHBL05, OTM+07, DE10,HXB03,Gre06,LWA06] have studied the causes of this fact and have identified some requirements for successful CDSS:

1. Clinical decision support should be computer based [KHBL05], as the classical formalizations of decision support recommendations (i.e. medical books and journals, specialized conferences, clinical guidelines and protocols) require a high processing effort that can not be assumed during daily practice.

2. CDSS should be integrated into the clinical workflow [HXB03,KHBL05,DE10, BKW+03] (i) to avoid the duplication of data introduction to users, and (ii) to provide support during all different tasks involved in daily practice.

3. CDSS should be easy to maintain and to extend, when new knowledge comes to the system [PT06, BL07], as classical maintenance requires a high effort to medical organizations.

4. Clinical decision support should be provided at the place and time when it is needed [HXB03, PT06, BKW+03].

5. Costs and effects of the implementation of CDSS in real clinical environments should be measured and evaluated [PT06].

6. An architecture that allows the sharing and reusing CDSS modules and services [SWO+08] should be developed.

The motivation for this PhD work is to show how semantics and experience-based technologies can enhance CDSS in order to cover the aforesaid requirements. Particularly, we believe that the use of semantic and experience-based technologies can improve CDSS in the following aspects:

- The use of semantic and experience-based technologies provides improved medical knowledge handling and reutilization: Because the knowledge sharing and reusing nature of semantic technologies will facilitate the gathering of the relevant knowledge during decision making processes.

- The use of semantic and experience-based technologies can reduce human error: Because providing physicians recommendations endorsed by bibliographic evidence and the experiences of the whole medical team can lead them to better decisions, that fit better the state and needs of the patients.

- The use of semantic and experience-based technologies can reduce health cost: Because providing physicians with tools that summarize the relevant data and knowledge for a certain decision in a unique page, speeds up the decision making process.

1.1.1 Problem Statement - Research question

The research question that motivates this work is:

> How can clinical experience be modeled, acquired and reused in the context of clinical decision making? Is it possible to develop a semantic steered clinical decision support system that allows the handling of the colective experience of a medical organization?

1.2 Thesis objectives

The objectives of this Thesis are:

- To review the most important concepts related to experience and decisional modeling.
- To review the most important concepts related to the state of art of current CDSS.
- To propose a methodology for the generation of the underlying ontologies and rules of a CDSS.
- To propose a methodology for the recommendations generation process.
- To propose a methodology for the automatic evolution of a ruleset, based on the acquired decicional events.
- To present a generic model for clinial tasks in the context of clinical decision making.
- To present a generic architecture for S-CDSS that fits in the clinical task model.
- To present a framework for the management of the clinical experience of a medical organization.

1.3 Summary of the main contributions

The methodological and technical contributions of this Thesis are:

1.3.1 Methodological contributions

- A methodology for the recommendations generation process is proposed and specified.
- A methodology for extending Reflexive Ontologies.

- A methodology for the automatic evolution of a ruleset based on the acquired decisional events is proposed and specified.
- A generic model for clinical tasks, called the Clinical Task Model (CTM) is proposed.
- A generic architecture Semantic steered Clinical Decision Support Systems (S-CDSS) that fits in the CTM is presented, allowing the management of the clinical experience of a medical organization.

1.3.2 Technical contributions

- A methodology for the generation of a domain ontology and a ruleset of a S-CDSS is presented and implemented in two domains: the early diagnosis of AD and the diagnosis and treatment of Breast Cancer.
- An implementation of a CDSS for the early diagnosis of AD is presented.
- Evaluation results of the implemented CDSS for the early diagnosis of AD are presented, with a deep focus on the measurement of the benefits provided by the implementation of Reflexive Ontologies.
- An implementation of a S-CDSS for the diagnosis and treatment on Breast Cancer is presented.
- An evaluation methodology of the implemented S-CDSS for the diagnosis and treatment on Breast Cancer is proposed.

1.4 Research environment and context where this Thesis was developed

The research and scientific contributions in this Thesis were generated during the participation of the PhD candidate in different research projects at the applied research centre Vicomtech-IK4 (San Sebastian, Spain), with the collaboration and guidance of Prof. Manuel Graña (University of the Basque Country UPV/EHU) and Dr. Carlos Toro (Vicomtech-IK4).

1.4.1 Endorsing projects

The most relevant projects that constitute the research framework of this thesis are briefly introduced below.

- The project MIND, *Multidisciplinary Approach to Alzheimer's Disease,* a Spanish national basic research project co-funded by the Centre for Industrial Technological Development (CDTI) and the Ministry of Economy and Competitiveness of Spain.

- The project LIFE, *Breast Cancer Challenge*, a Spanish national research project co-funded by the CDTI and the Technology Fund of the European Union FEDER funds.

1.4.2 Endorsing publications

Submitted

1* **Sanchez, E**; Wang, P; Toro, C; Sanin, C; Graña, M; Szczerbicki, E; Artetxe, A; Carrasco, E; Guijarro, F; Brualla, L: Decisional DNA for modeling and reuse of experiential-based clinical assessments in breast cancer diagnosis and treatment. In: Neurocomputing (2014).

2 Mesa, I; **Sanchez, E**; Toro, C; Diaz, J; Artetxe, A; Graña, M; Guijarro, F; Martinez, C; Jimenez, JM; Rajasekharan, S; Alarcon, JA; De Mauro, A: Design and development of a mobile cardiac rehabilitation system. In: Cybernetics and Systems (2014). (*Accepted for publication*)

Published

3* **Sanchez, E**; Toro, C; Artetxe, A; Graña, M; Sanin, C; Szczerbicki, E; Carrasco, E; Guijarro, F: Bridging challenges of clinical decision support systems with a semantic approach. A case study on breast cancer. Pattern Recognition Letters, Volume 34, Issue 14, pp. 1758-1768 (2013).

4 Mesa, I; **Sanchez, E**; Diaz, J; Toro, C; Artetxe, A; Graña, M; Guijarro, F; Martínez, C; Jiménez, JM; Alarcon, JA; De Mauro, A: GoCardio – A novel approach for mobility in cardiac monitoring. In: Howlett, RJ; Tsihrintzis, G; Toro, C; Virvou, M; Jain, L (Eds) Innovation in Medicine and Healthcare, pp. 110-120 (2013).

5 Artetxe, A; **Sanchez, E**; Toro, C; Sanín, C; Szczerbicki, E; Graña, M; Posada, J: Impact of Reflexive Ontologies in Semantic Clinical Decision Support Systems. In: Cybernetics and Systems Volume 44, Issue 2-3, pp. 187-203 (2013).

6 Toro, C; **Sanchez, E**; Carrasco, E; Mancilla-Amaya, L; Sanín, C; Szczerbicki, E; Graña, M; Bonachela, P; Parra, C; Bueno, G; Guijarro, F: Using Set of Experience Knowledge Structure to Extend a Rule Set of Clinical Decision Support System for Alzheimer's Disease Diagnosis. In: Cybernetics and Systems Volume 43, Issue 2, pp. 81-95 (2012).

7 Sanín, C; Toro, C; Zhang, H; **Sanchez, E**; Szczerbicki, E; Carrasco, E; Wang, P; Mancilla-Amaya, L: Decisional DNA: A multi-technology shareable knowledge structure for decisional experience. In: Nguyen, NT; Jędrzejowicz, P; Lee, G (Eds) Neurocomputing Volume 88, pp. 42-53 (2012).

8* Sanchez, E; Toro, C; Artetxe, A; Graña, M; Carrasco, E; Guijarro, F: A Semantic Clinical Decision Support System: conceptual architecture and implementation guidelines. In: Graña, M; Toro, C; Posada, J; Howlett, RJ; Jain, LC (Eds) Frontiers in Artificial Intelligence and Applications, Volume 243: Advances in Knowledge-Based and Intelligent Information and Engineering Systems, pp. 1390-1399. IOS Press (2012).

9 Artetxe, A; **Sanchez, E**; Toro, C; Sanín, C; Szczerbicki, E; Graña, M; Posada, J: Speed-up of a Knowledge-Based Clinical Diagnosis System using Reflexive Ontologies. In: Graña, M; Toro, C; Posada, J; Howlett, RJ; Jain, LC (Eds) Frontiers in Artificial Intelligence and Applications, Volume 243: Advances in

Knowledge-Based and Intelligent Information and Engineering Systems, pp. 1480-1489, IOS Press (2012).

10* **Sanchez, E**; Toro, C; Carrasco, E; Bueno, G; Parra, C; Bonachela, P; Graña, M; Guijarro, F: An Architecture for the Semantic Enhancement of Clinical Decision Support Systems. In: König, A; Dengel, A; Hinkelmann, K; Kise, K; Howlett, RJ; Jain, LC (Eds) Knowlege-Based and Intelligent Information and Engineering Systems, Lecture Notes in Computer Science Volume 6882, II, pp. 611-620, Springer Berlin Heidelberg (2011).

11* **Sanchez, E**; Toro, C; Carrasco, E; Bonachela, P; Parra, C; Bueno, G; Guijarro, F: A Knowledge-based Clinical Decision Support System for the diagnosis of Alzheimer Disease. In: Proceedings of the 13th IEEE International Conference on e-Health Networking, Application & Services (Healthcom 2011), Columbia, Missouri, USA. 13 June, pp. 355-361, (2011).

Published (Poster presentations)

12 González Sanchis, A; Brualla, L; Gordo Partearrollo JC; Ferrer, J; Leal, A; Ugarriza, A; **Sanchez, E**; Fuster, C; Sanchez Carazo, J; Estornell, J; Roselló, J; López Torrecilla, J; Belloch, V: Breast cancer integral challenge: Towards a personalized medicine. In: Reports of Practical Oncology & Radiotherapy Volume 18, Supplement 1, pp 167-168, (2013).

13 González Sanchis, A; Brualla, L; Gordo Partearrollo JC; Ugarriza, A; **Sanchez, E**; Ferrer, J; Fuster, C; Roselló, J; López Torrecilla, J; Belloch, V: Computer support to optimize decisions in breast functional units. In: Reports of Practical Oncology & Radiotherapy Volume 18, Supplement 1, pp 170, (2013).

14 González Sanchis, A; Ferrer, J; Brualla, L; Gordo Partearrollo JC; Sanchez Jurado, R; Cozar Santiago, MP; **Sanchez, E**; Ugarriza, A; Roselló, J; López Torrecilla, J; Rubio, D: Soporte informático para el diagnóstico y tratamiento individualizado del Cáncer de Mama en las Unidades Funcionales de Mama. In: Proceedings of the 1st Spanish Congress on Breast Cancer 17-19 October,

Madrid, Spain (Libro de Ponencias y Publicaciones del 1er Congreso Español de la Mama, ISBN 978-84-941728-3-0), (2013) (*in Spanish*).

15 González Sanchis, A; Brualla, L; Gordo Partearrollo JC; Ferrer, J; Ugarriza, A; Leal, A; **Sanchez, E**; Fuster, C; Sanchez Carazo, J; Estornell, J; Garcia, R; Roselló, J; López Torrecilla, J; Belloch, V: Medicina personalizada e integral en el Cáncer de Mama. In: Proceedings of the 1st Spanish Congress on Breast Cancer 17-19 October, Madrid, Spain (Libro de Ponencias y Publicaciones del 1er Congreso Español de la Mama, ISBN 978-84-941728-3-0), (2013) (*in Spanish*).

1.5 Structure of this Thesis

This Thesis is structured as follows: First we will present a state of the art, containing the basic technologies and concepts on which this work is based. Such technologies are divided in two groups, (*i*) experience and decisional modeling, presented in Chapter 2, and (*ii*) CDSS, presented in Chapter 3. Following the presentation of such technologies, we will introduce our contributions to Semantic steered Clinical Decision Support Systems. Such contributions will be divided in two, (*i*) the Methodological Contributions and (*ii*) the Technical Contributions. Figure 1.1 depicts the aforementioned structure.

Methodological contributions to Semantic Clinical Decision Support Systems

- **Chapter 4 - Reasoning and recommendation generation**

  In this Chapter we will present the reasoning process for the generation of recommendations provided by a CDSS. We will specify the elements involved in the process, such as the ontology, the queries and the ruleset. In order to speed up the reasoning process we will propose the application of the Reflexive Ontologies technique, where the queries are contained in the ontology. Additionally we propose the Extended Reflexive Ontologies, where rules are also contained in the ontology, for a higher speed up of the reasoning process for recommendations generation. We will describe the process in the three cases (with plain ontologies, Reflexive Ontologies and Extended Reflexive Ontologies).

- **Chapter 5 - Experience-based learning**

  In this Chapter we will present the experience-based learning approach that supports knowledge maintenance of the CDSS. We will apply SOEKS/DDNA technologies for the construction of the basic experience data estructure of our approach. We will explain the experience acquisition and consolidation process of the system, based on such structure. In particular, three different algorithms will be detailed for the evolution of the ruleset of the system: an algorithm for rule weight evolution, an algorithm for fine-tuning of rules and a new rule generation algorithm.

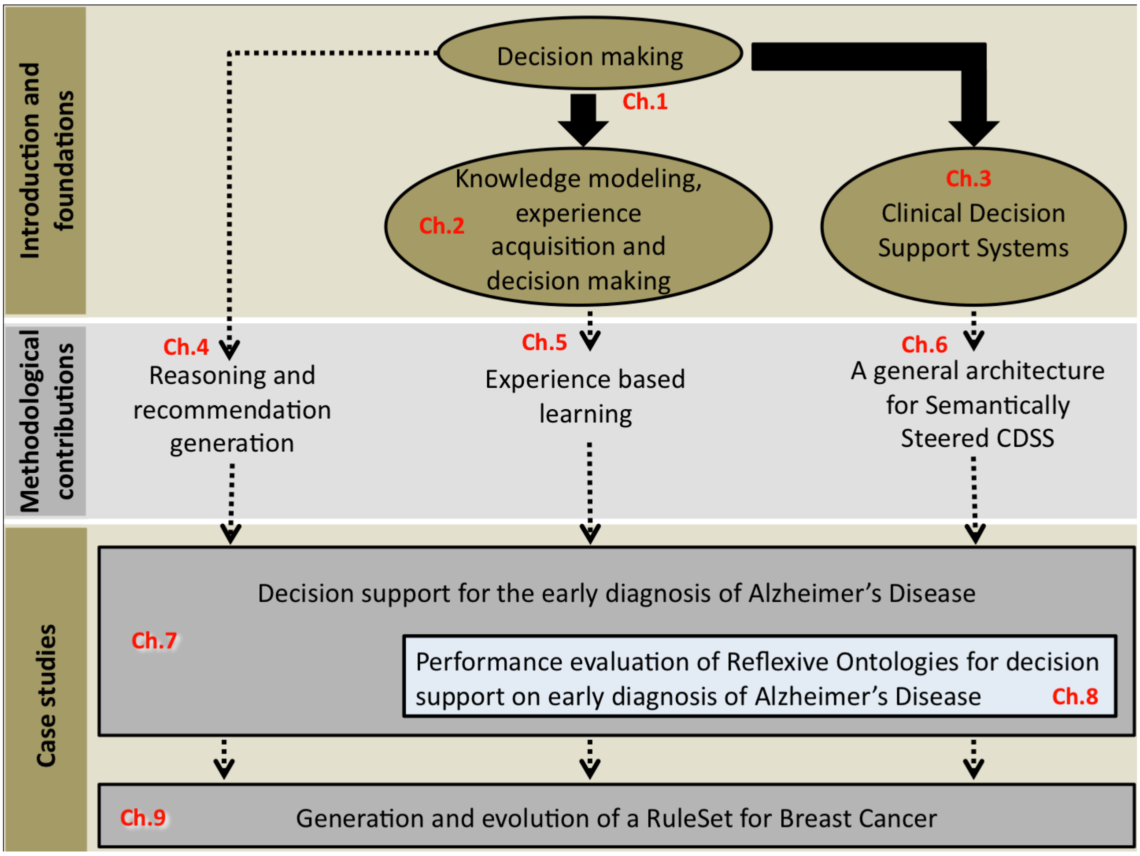

Figure 1.1: Structure of the Thesis

- **Chapter 6 - A general architecture for Semantically Steered CDSS (S-CDSS)**

  In this Chapter we will present an extensive and modular architecture for CDSS, oriented to covering the aforesaid requirements. We call it Semantic CDSS (S-CDSS), as it is based on semantic and experience based tecnologies. As the architecture is oriented to covering the whole clinical workflow we first introduce the Clinical Task Model (CTM), where the different clinical task are modeled. Our architecture for S-CDSS fits into the CTM.

Technical contributions to Semantic Clinical Decision Support Systems

Chapters 7 and 8 will introduce different experiences related to semantic CDSS that were undertaken/developed during the work in different R&D projects. Such presentation is divided as follows:

- **Chapter 7 - Decision support for the early diagnosis of Alzheimer's Disease (AD)**

  In this Chapter a CDSS for the early diagnosis of AD is presented, developed under the framework of the MIND research project. We present the MIND Ontology, a domain ontology we developed for the diagnosis of Alzheimer Disease, as well as the mappings of the MIND Ontology with other standard ontologies. We also present the implementation of a set of production rules for the early diagnosis of AD we developed. Finally, we detail the implementation of the recommendations generation process of Chapter 4, based on Reflexive Ontologies.

- **Chapter 8 - Performance evaluation of Reflexive Ontologies for decision support on early diagnosis of Alzheimer's Disease**

  In this Chapter we present an empirical evaluation of the performance of Reflexive Ontologies (RO). We evaluate the speed up provided by RO in an specific case (not generalizable): the implementation of the MIND CDSS, presented in Chapter 7. We present the methodology, test environment and results obtained from the evaluation performed.

- **Chapter 9 - Generation and evolution of a RuleSet for Breast Cancer**

  In this Chapter we present a S-CDSS for the diagnosis and treatment of Breast Cancer. We first present the Life Ontology, a domain ontology we developed for Breast Cancer, its mappings to standard ontologies and a set of production rules for Breast Cancer diagnosis and treatment. We then present implementation of the S-CDSS architecture. Finally, the implementation of the experience acquisition and consolidation process and the rule evolution algorithms are detailed.

Finally in **Chapter 10**, we will present the conclusions of this Thesis and propose some future work.

Chapter 2

Knowledge modeling, experience acquisition and decision making

This Chapter introduces the reader into the most relevant concepts of this Thesis. The majority of our work is based on knowledge modeling and computational semantics. In this Chapter we introduce the process of decision making, presenting a comprenhensive state of the art. Our focus is on a specific family of techniques which emphasize the role of experience in knowledge building. Amongst the vast panorama of decision making techniques proposed in Artificial Intelligence, we centered most of our contributions on the *Set of Experience Knowledge Structure* (SOEKS) [CS09] which provides a framework for the representation of experience as an asset. This interesting approach leads to new knowledge structures managed by *Decisional DNA* (DDNA) techniques [CS07] also presented in the literature and applied to different domains with good sucess.

This Chapter is structured as follows: Section 2.1 gives an introduction of domain modeling, knowledge representation, and semantic reasoning. Section 2.2 contains an introduction to decision making, a brief overview of technologies, and an introduction to decisional experience. Section 2.3 introduces experience handling and the context where SOEKS and DDNA where proposed. Finally, Section 2.4 describes SOEKS and DDNA technologies.

2.1 Knowledge representation and reasoning

Knowledge Engineering (KE) is an engineering discipline in which knowledge is incorporated into computer systems with the aim of solving complex problems that generally require a high human expertise [FM83]. KE involves different techniques for representing, modeling and the reuse of knowledge that are broadly reviewed in this section.

When modeling a particular domain, for instance, an agreed vocabulary for the description of the different elements of the domain can be achieved, as well as the relationships between elements. Such domain model can be represented in different languages, depending on the expressiveness and semantics needed on further reasoning tasks over the knowledge. In [SKDO06] different types of domain models are considered, i.e. taxonomies, thesauri or ontologies. The three of them classify relationships between elements, but have different purposes and provide benefits at different levels:

(i) A taxonomy stablishes a hierarchy between the different concepts of the domain (not associational nor equivalence relationships), it is adequate for

(ii) A thesaurus stablishes the structure of concepts of a domain and their relationships, which can be hierarchical, associational and equivalence.

(iii) An ontology is a formal representation of knowledge as a set of concepts and their interelationships within a domain, where agreements are reached to use a vocabulary in a coherent and consistent manner. It defines the domain in great detail, as relationships are explicitly typed, the properties and value types of concepts are defined, and the instances of such concepts and relationships are also contained.

In our work, we have modeled the domain with ontologies. On the following sections, a brief introduction to the most relevant aspects of ontologies are reported.

2.1.1 Ontological engineering

The root onto- comes from the Greek ὤν, ὄντος, ("being", "that which is") and the term ontology has its origin in philosophy. However, among history ontologies have been studied in different context and in this Thesis we will focus on the application of ontologies into computer science. In the latter the most widely accepted ontology definition is the one by Gruber as the explicit specification of a conceptualization [Gru95]. Guarino refined such definition, by stating that an ontology is the explicit and partial account of a conceptualization [GG95].

2.1.1.1 Upper and domain ontologies

Depending on the domain of application or a generalistic "ontology of the world approach" , ontologies can be designed within two very different perpectives:

Foundational ontologies Foundational ontologies, also known as upper or top-level, are a model of the common objects applicable across a wide range of domains and developed to characterize explicitly a viewpoint of a reality [BL04]. They are built upon a core vocabulary that contains the terms and associated object descriptions as they are used in various relevant domain sets. The most relevant foundational ontologies are GFO [HH06], SUMO [OAH+07] and DOLCE [OAH+07], amongst others.

Domain ontologies Domain ontologies describe a set of representational primitives that model a domain of knowledge or discourse, providing a common and unambiguous understanding of a domain for both the users and the system [ZHCZ13]. It models a specific domain, without any pretension of generality. For instance, in an ontology of industrial manufacturing, a screw can be a manufactured part, but instead, in a furniture ontology, an assembly tool.

Figure 2.1: Portion of the SUMO ontology hierarchy

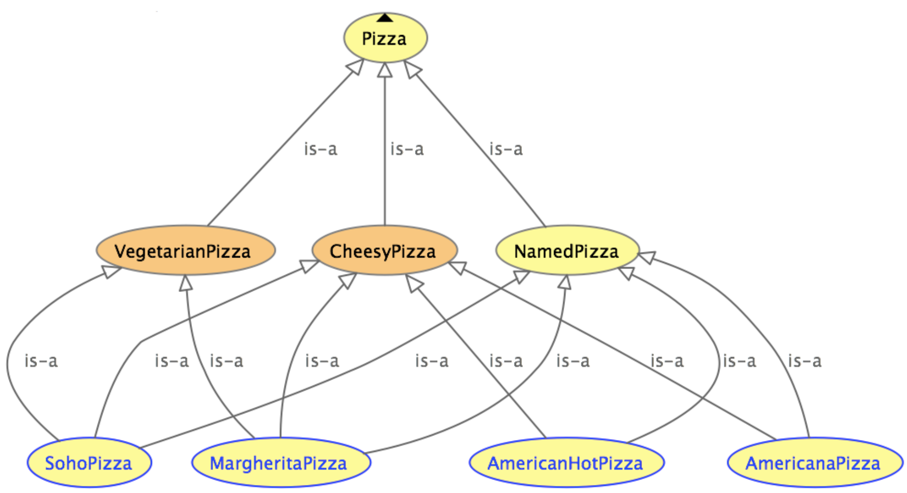

Figure 2.2: Portion of the subclass hierarchy of the pizza ontology [Hor11]

2.1.2 Ontology components

Ontologies are composed by eight different elements:

Individuals: atomic instances corresponding to the basic elements of an ontology,

Classes: sets of instances which described as collections, concepts, classes in programming, types of objects, or kinds of things depending on the domain of application; they are usually built by specifying some properties.

Attributes: variables that describe quantitative or qualitatively aspects, properties, features, characteristics, or parameters of objects in a classes, basically, objects in the same class are described by the same variables.

Relations: they specify ways in which classes and individuals can be related to one another.

Functions: representing complex structures formed from certain relations that can be used in place of an individual term in a statement.

Restrictions: formally stated descriptions of clauses defined on the input that must be true in order for the reasoning to work upon it.

Rules: statements in the form of an if-then-else (antecedent-consequents) sentence that describe the logical inferences that can be drawn from an assertion]in a particular form.

Axioms: assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application.

2.1.3 OWL ontology language

An ontology language is a formal language used to encode the ontology [AH03].

OWL is a language for making ontological statements, developed as a follow-on from RDF and RDFS, as well as earlier ontology language projects including OIL, DAML, and DAML+OIL. OWL is intended to be used over the World Wide Web, and all its elements (classes, properties and individuals) are defined as RDF resources, and identified by URIs.

OWL is the most recent development in standard ontology languages, endorsed by the World Wide Web Consortium (W3C) to promote the Semantic Web vision.

"An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms"[1].

2.1.4 Ontology development methodologies

Motivation for building ontologies

We would like to recall the reader the reasons for developing ontologies as presented by Noy et al [NM01]. Ontologies are useful because they provide means for:

(i) Allowing the communication between people and/or software agents, by sharing a common model of the structure of information. In a classical example, if several different Web sites containing information on the same domain share the same underlying ontology of the terms used, then computer agents will be able to extract and aggregate information from them, in order to answer user queries or input data to other applications.

(ii) Allowing reuse of domain knowledge, so that no duplicate efforts need to be made when modeling knowledge. If one group of researchers develops an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large ontology, we can integrate several existing ontologies describing portions of the large domain. We can also reuse a general ontology, and extend it to describe our domain of interest.

(iii) Allowing flexibility in software development, by making it possible to surface domain assumptions underlying an implementation and change them easily when our knowledge about the domain changes. Programming language coding buries and freezes assumptions about the world in the code, so that these assumptions become not only hard to find and understand, but also hard to change, in particular for someone without programming expertise.

[1]OWL 2 Specification: `goo.gl/gZmcOR`

(iv) Allowing clear separation of domain knowledge from operational knowledge. We can configure a product from its components according to a required specification and implement a program that does this configuration independent of the products and components themselves.

(v) Analyzing domain knowledge is possible once a declarative specification of the terms is available. Such analysis can be extremely valuable reutilization and extension of existing ontologies.

(vi) Allowing other programs to use the set of data and their structrure defined by the ontology.

Methodologies for ontology development

There are quite a few ontology development methodologies. In this Section, we will concentrate in two of them, which were relevant for our work.

Ontology Development 101 [NM01], based on seven steps:

1. Determine the domain and scope of the ontology
2. Consider reusing existing ontologies
3. Enumerate important terms in the ontology
4. Define the classes and the class hierarchy
5. Define the properties of classes—slots
6. Define the facets of the slots (Slot cardinality; Slot-value type; Domain and range of a slot)
7. Create instances

Methontology [GPFC04], based on eleven steps:

1. Build the glossary that identifies the set of terms to be included in the ontology, their natural language definition, and their synonyms and acronyms.

2. Build concept taxonomies to classify concepts. The output of this task will be one or more taxonomies where concepts are classified.

3. Build *ad hoc* binary relation diagrams to identify *ad hoc* relationships between concepts of the ontology and with concepts of other ontologies.

4. Build the concept dictionary, which mainly includes the concept instances for each concept, their instance and class attributes, and their *ad hoc* relations.

5. Describe in detail each *ad hoc* binary relation that appears on the *ad hoc* binary relation diagram and on the concept dictionary. The result of this task is the *ad hoc* binary relation table.

6. Describe in detail each instance attribute that appears on the concept dictionary. The result of this task is the table where instance attributes are described.

7. Describe in detail each class attribute that appears on the concept dictionary. The result of this task is the table where class attributes are described.

8. Describe in detail each constant and to produce a constant table. Constants specify information related to the domain of knowledge, they always take the same value, and are normally used in formulas.

9. Describe formal axiom that are used for constraint checking.

10. Describe rules that are used for inferring values for attributes.

11. Introduce information about instances.

2.1.5 Ontology editors

Ontology editors are applications designed to assist in the creation or manipulation of ontologies. They often express ontologies in one of many ontology languages. There are several ontology editors, such as HOZO, JOE, FluentEditor for OWL, etc.

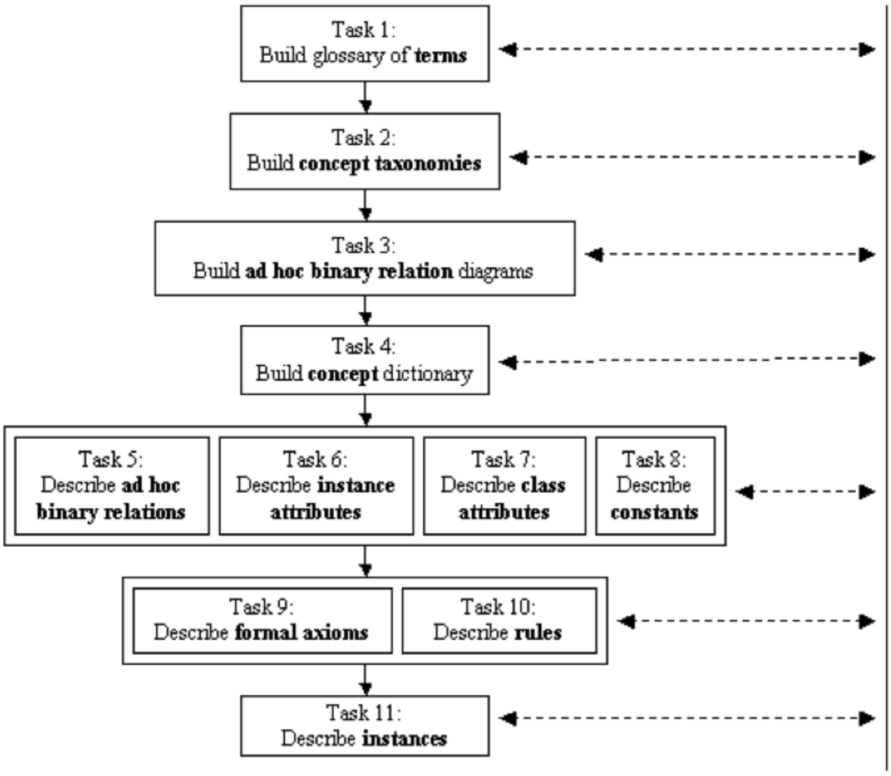

Figure 2.3: Diagram of the tasks performed in Methontology [GPFC04]

In this Thesis we have used Protégé-OWL[2]. The Protégé-OWL editor is an extension of Protégé that supports the Web Ontology Language (OWL). Protégé is a US national resource for biomedical ontologies and knowledge bases supported by the National Institute of General Medical Sciences and a core component of The National Center for Biomedical Ontology, developed by Stanford Center for Biomedical Informatics Research.

The Protégé-OWL editor enables users to:

1. Load and save OWL and RDF ontologies.

2. Edit and visualize classes, properties, and SWRL rules.

3. Define logical class characteristics as OWL expressions.

4. Execute reasoners such as description logic classifiers.

5. Edit OWL individuals for Semantic Web markup.

[2]Protégé-OWL web page: `goo.gl/NmGirH`

Protégé-OWL is tightly integrated with Jena[3] and has an open-source Java API for the development of custom-tailored user interface components or arbitrary Semantic Web services.

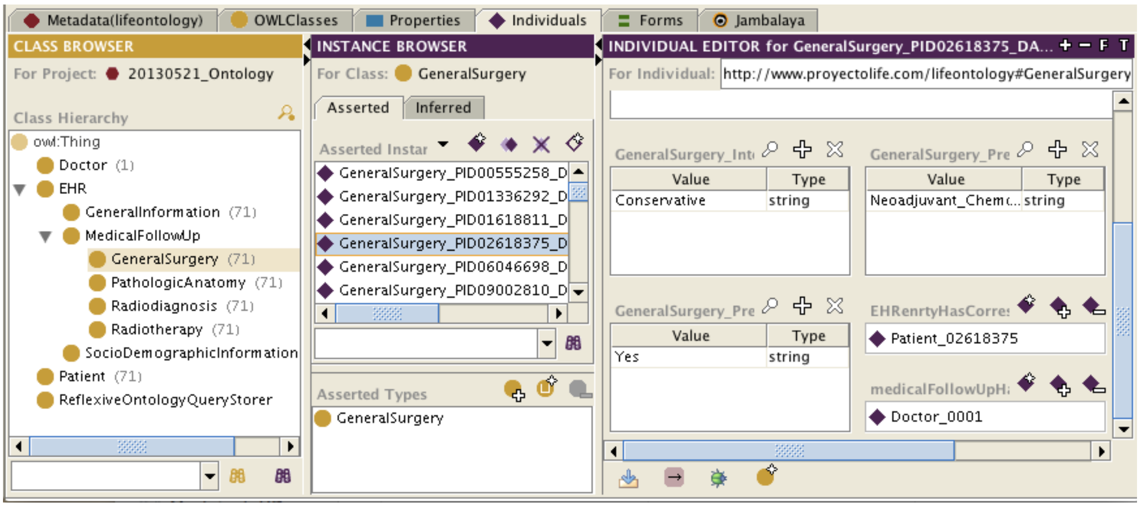

Figure 2.4: Example screenshot of the Protégé-OWL ontology editor

2.1.6 Ontology repositories

Ontological engineering is oriented to share and reuse existig ontologies, as far as possible. With such purpose, ontology libraries or repositories have been generated and offered in the web as open services. The most relevant are:

- The Open Biological and Biomedical Ontologies (OBO) Foundry is a collaborative experiment involving developers of science-based ontologies who are establishing a set of principles for ontology development with the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain[4].

- Bioportal (ontology repository of NCBO) [NSW+09] is an open repository of biomedical ontologies that via Web services and browsers provides access (i.e. browse, search and visualize) to ontologies, developed in different languages and formats, such as OWL, RDF, OBO format, Protégé frames)[5]. Ontology

[3]Jena Home Page: `goo.gl/Zru2Ct`
[4]OBO Foundry Home page: `goo.gl/WuIIn8`
[5]BioPortal Home page: `goo.gl/g9qe11`

reviewing is also supported by BioPortal, in order to provide a social framework to further develop and reuse existing ontologies.

2.2 Decision making

In this section we describe the relevant concepts regarding decision making, which later underly clinical decision systems.

2.2.1 Definitions and background concepts

The process of clinical decision-making was not studied in deep until the late 1990's, when concerns on avoiding medical errors arose. Before that moment, clinical decision-making was accepted as an esoteric matter that only concerned physicians, and no effort was done to analyze possible errors and their causes [CN11]. During these last years, the underlying processes have been studied from the psychological point view, and a consensus has been reached to establish the Dual Process Theory as a valid and robust approach [CN11,ES02,Els09,MSE08,Pre08]. The Dual Process Theory differentiates two types of clinical decision making [CN11]:

Analytical, consisting of testing hypotheses and concluding the most likely one [ES02]. It is focused on scientific rigor following a rule-based approach (deductive reasoning). Nevertheless, this approach requires a high reasoning effort to doctors, and the literature reports [Els09, MSE08, Pre08, SSF+11] that they mostly follow a more intuitive clinical reasoning, as described below.

Intuitive, based on previous experiences on similar situations [RR00,Cio01]. This approach requires a much lower effort to decision makers, but is subject to a higher error rate [CN11]. In fact, if the input case is not correctly identified and the similarity process with prior experiences does not take into account all relevant parameters, the final decision may not be adecuate. In general, everyday clinical practice is very influenced by previous experiences. This fact is reflected in medical learning and training programs, which emphasize the acquisition of new experiences to enrich the knowledge of trainees [SSF$^+$11, TDE$^+$12, Kas10].

Decision systems based on symbolic approaches apply reasoning processes similar to those followed by physicians in real life [AWP05]. Actually, recent literature in CDSS presents approaches following both reasoning types. We hypothesize that, in the same manner as with physicians, a combined approach of analytic and intuitive modes for the CDSS reasoning processes could support both the production of recommendations and the update of the knowledge in the system. Thus, we follow a mixed approach were (i) recommendations are generated based on a set of production rules given by medical experts and (ii) those rules are updated by the system with the acquired experience.

2.2.2 A quick sample of non-knowledge based Artificial Intelligence techniques

For the sake of completitude, we would like to review some classsification approaches that have been used in the literature to build CDSS that are not directly in the knowldge domain (let us call them classical approaches). The following techniques are mostly on a data/information level and for such reasons were not directly used within our approach. Furthermore, these apporaches are very much focused on specific quantitative data (not a general clinical decision making process).

Support Vector Machines In Support Vector Machines (SVM) [Vap98] the set of support vectors allows to define the discriminating surface providing the greatest separation (margin) between the classes. That is, the decision function can be expressed in terms of the support vectors only. Their theoretical and application qualities [Vap98] have attracted attention from the pattern recognition community [FS07, TTLW06]. More precisely, SVM separates a given set of two-class labelled training data into two subspaces by a hyperplane which is maximally distant from the two classes (see Figure 2.5). Such maximal margin hyperplane will provide the greatest generalization of the classifier to unseen new data samples. If the linear separation of the training data is not possible, the kernel trick allows SVMs to build a non-linear decision boundary [Vap98]. The learning of the discrimination function is achieved by the dual minimization of a cost function involving the classification error and some ad hoc regularization parameters. SVM have been adopted by the neuroscience community as the stardard classifier, specially in some neuroimage classification applications. Parameter tuning for SVM is critical to obtain good performance. Thus, each learning experiment needs to be wrapped with a grid search procedure [LT$^+$09].

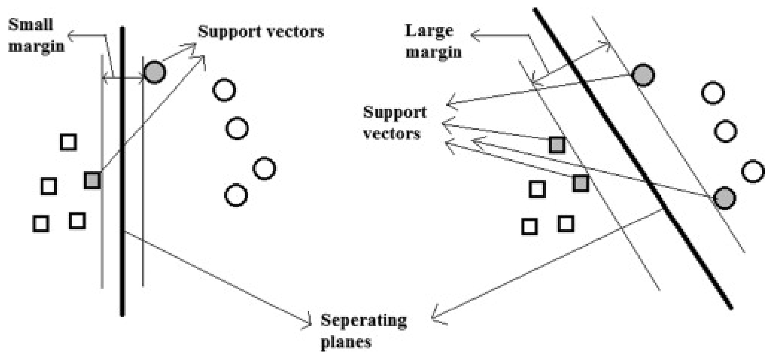

Figure 2.5: Support Vector Machine hyperplane for the separation of two classes [NJR14]

Neural Networks

Artificial Neural Networks [RHW86,Hay98,HDB95] introduced quantitative training algorithms following a biological inspiration, thus revolutionizing the field of Artificial Intelligence by shifting from logic based reasoning approaches to a numerical and statistics paradigm, which allows quantitative computational experiments for validation of the learned systems. In Figure 2.6 the architecture of a typical Neural Network is shown.

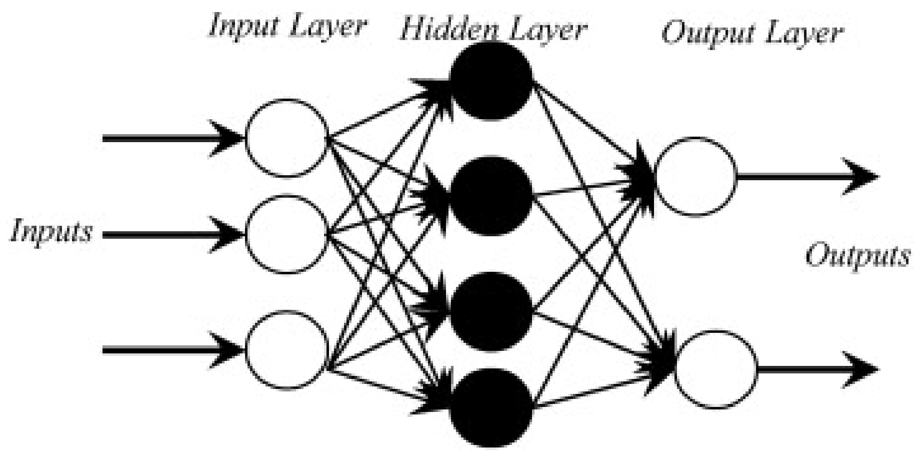

Figure 2.6: Classical architecture of a Neural Networks (multilayer perceptron) [TA14]

Multi Layer Perceptron trained with Backpropagation Backpropagation of errors (BP) [RHW86, Hay98, HDB95] is a non-linear generalization of the squared error gradient descent learning rule for learning the weights of single-layer perceptron, allowing to extend it to feed-forward network architectures, aka Multi-Layer Perceptron (MLP), despite the fact that they do not fit into the computational framework of the Perceptron. BP requires that the activation function used by the artificial neurons (or "nodes") is differentiable with its derivative being a simple function of itself. The backpropagation of the error allows to compute the gradient of the error function relative to the hidden units. It is analytically derived using the chain rule of calculus. During on-line learning, the weights of the network are updated at each input data item presentation. The MLP trained with BP has been used in many applications, in fact its introduction in the 80s was a strong revulsive in the Artificial Intelligence landscape.

Radial Basis Function Networks Radial Basis Function networks (RBF) [CCG91, Hay98] are artificial neural networks whose neuron activation function is a radial basis function. RBFs consist of a two layer neural network, where each hidden unit implements a radial activated function. The output units compute a weighted sum of hidden unit outputs. Training consists of an unsupervised training of the hidden units followed by the supervised training of the output units weights. They posses strong approximation properties that have favored their application in many fields, mostly in control tasks.

Probabilistic Neural Networks A Probabilistic Neural Network (PNN) [Spe90] is a special type of neural network that uses a kernel-based approximation to build an estimate of the probability density function of each class in a classification problem. The distance from the each sample point to each of the remaining sample points is computed. A *radial basis function* (RBF) is applied to the distance to compute the weight (influence) for each point. Different types of radial basis functions could be used, but the most common is the Gaussian function. Its sigma parameter determines the *spread* of the RBF function; that is, how quickly the function declines as the distance increased from the point. With larger sigma values the function has more spread, so that distant points have a greater mutual influence. PNN is a kind of 1-NN classifier that uses all the training data samples as reference values and the only functional transformation is the computation of the posterior probability of the classes as a combination (sum/average) of the evidence given by each data sample through its RBF window. The tuning of a PNN network looks for the optimal sigma value of the spread of the RBF functions.

Learning Vector Quantization Learning vector quantization (LVQ) [Koh89, SK99] provides a method for training competitive networks in a supervised manner. The system is composed of an unsupervisedly trained competitive layer which performs a partitioning of the input space. The supervisedly trained output layer provides the labeling of the input data according to its belonging to an input region (crisp clustering) or to its degree of membership (soft clustering). In the original proposition of the LVQ, the competitive units were cluster centres with the Euclidean distance as the similitude measure. Training of the competitive units can be performed by Kohonen's Self Organizing Map. Supervised training was simply the assignment of a label to a competitive unit according to a majority voting on the data samples falling in the partition corresponding to the unit. LVQ provides fine tuning of the competitive units using class information.

Random Forests (RF)

The random forests (RF) algorithm is a classifier [Bre01] that encompasses bagging [Bre96] and random decision trees [AG97, Ho98]. RF became popular due to its simplicity of training and tuning while offering a similar performance to boosting. Consider a RF collection of tree predictors, that is, a RF is a large collection of decorrelated decision trees, where each tree casts a unit vote for the most popular class of the input. RF capture complex interaction structures in data, and are supposed to be resistant to over-fitting of data if individual trees are sufficiently deep.

Extreme Learning Machines

Extreme Learning Machines (ELM) [HZS06] is a fast training approach for the single layer feed-forward neural networks (SFLN), providing on average good quality classification and regression results whose computation burden is orders of magnitude lower than conventional backpropagation training. This method is based on the Moore-Penrose generalized inverse providing the minimum Least-Squares solution of general linear systems. In the short time since their proposal, ELMs have been successfully applied to a large number of problems such as face recognition [MG12]. However, an ELM major criticism is the uncertainty of its performance due the random generation of the hidden layer weights. An approach to overcome that problem is the composition of elementary classifiers into ensembles, such as the Voting ELM (V-ELM) [CLHL12], which is a direct composition by majority voting of a collection of ELMs trained independently.

Case based reasoning

The intuitive computational decision approach is followed by systems based on Case-Based Reasoning [CM10a, dABW+98, HK11, WW04, ABF12, AP94]. Their major limitation is that the quality of the output depends on the previous cases included in the knowledge base. Systems following the analytic approach have also been proposed [KF11, KXY08], where recommendations are generated based on knowledge that is supported by medical literature and evidence. The weakness of these approaches is the difficulty for continuously updating the knowledge and decision criteria applied for the reasoning.

2.3 Experience handling

This Thesis has been heavily influenced by the experience based systems developed by Szerbicki and Sanin [CS07, CS09, SMASC09] and extended by [Tor08, TSC+12, STZ+12, SST07, TVS+07]. Experience, as a general concept, comprises previous knowledge or a skill obtained through daily life [SG03, SF04]. Usually experience is understood as a type of knowledge that one has gained from practice rather than books, research, and studies [SSG12]. In this way, experience or experiential knowledge can be regarded as a specialization of knowledge that includes information and strategies obtained from performing previous tasks. When these tasks involve making decisions, the specific experience that is gained is called decisional experience.

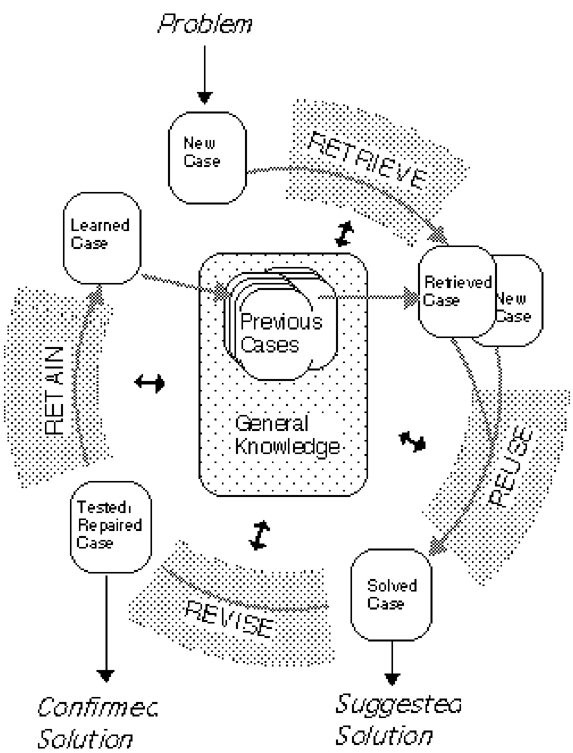

Figure 2.7: Case based reasoning cycle

The importance of decisional experience in knowledge engineering, and especially in knowledge sharing, has been recognised for at least last ten years. European and

Australian studies reported in [BD03] have established that the primary research aim of knowledge management (KM) should be to use the vast experience accumulating each day within organisations and systems, as far as true knowledge is developed through learning from current and past experiences [EA04, Ber04]. Experience management (EM), its formalization, representation, and experience based systems development is capturing increasingly growing attention of researchers and practitioners. However, the related problems and their solutions do not appear to have progressed too far. The fundamental limitation of current research in this area is that none of the proposed approaches uses experience as ongoing, real time reference during the decisional process in a way similar to what happens naturally when humans make decisions to answer a new situation. Existing techniques used to model experience, such as Case Base Reasoning [CGC13, Hül07], Decision Trees [SC09], Petri Nets [Zai13, DR05], and many others, lack the same critical element in order to assure useful real life implementations – they don't store and reuse experience in an ongoing, real-time representation system that can provide the following, crucial for useful decision support end user applications, features:

- Adaptability and cross-platform portability,
- Compactness and efficiency,
- Configurability and shareability,
- Security and trust, and
- Being exclusively experience oriented.

2.4 Set of Experience Knowledge Structure and Decisional DNA

Knowledge has been an important asset for individuals, organizations, and society through the ages. Decision makers, in general, base their current decisions on lessons learned from previous similar situations [CS09]; however, much of the experience held by individuals is not properly capitalized because of inappropriate knowledge representation or administration. This leads to decision reprocessing, inadequate response times, and lack of flexibility to adapt when new environment conditions are found. In order to represent and reuse experience in an adequately form, Sanin and Szczerbicki proposed the concepts of the Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA) [CS09, CS07, SMASC09].

The SOEKS knowledge representation of formal decision events in an explicit way, and it is based on four basic elements which are considered to be crucial in decision-making actions: Variables (V), Functions (F), Constraints (C), and Rules (R).

Variables are used to represent knowledge in an attribute-value form, following the traditional approach for knowledge representation. Given that the set of F, C, and R of the SOEKS are different ways of relating knowledge variables, it is safe to say that the latter are the central component of the entire knowledge structure.

Functions describe associations between a dependent variable and a set of input variables; therefore, the SOEKS uses functions as a way to establish links among variables and to construct multi-objective goals (i.e. multiple functions).

Constraints are functions that act as a way to limit possibilities, restrict the set of possible solutions, and control the performance of the system with respect to its goals.

Rules are used to represent inferences and correlate actions with the conditions under which they should be executed. Rules are relationships that operate in the universe of variables, and express the connection between a condition and a consequence in the form IF-THEN-ELSE.

A SOEKS is an molecular unit of knowledge acquisition through experience, where the decisional events are atomic units. The SOEKS is the basis for the creation of DDNA, which is a structure capable of capturing decisional fingerprints of an individual or organization. The name of Decisional DNA is an allegory to DNA because of its structure and the ability that if offers to store experience within itself. Let us ellaborate on this metaphor: the four elements that comprise a SOEKS can be compared to the four basic nucleotides of human DNA, and they are also connected in a way that resembles a human gene. A gene guides hereditary responses in living organisms, and analogously a SOEKS guides responses in decision-making processes. In that way, SOEKS is carried into the future by DDNA [SMASC09,CS09] as illustrated in Figure 2.8.

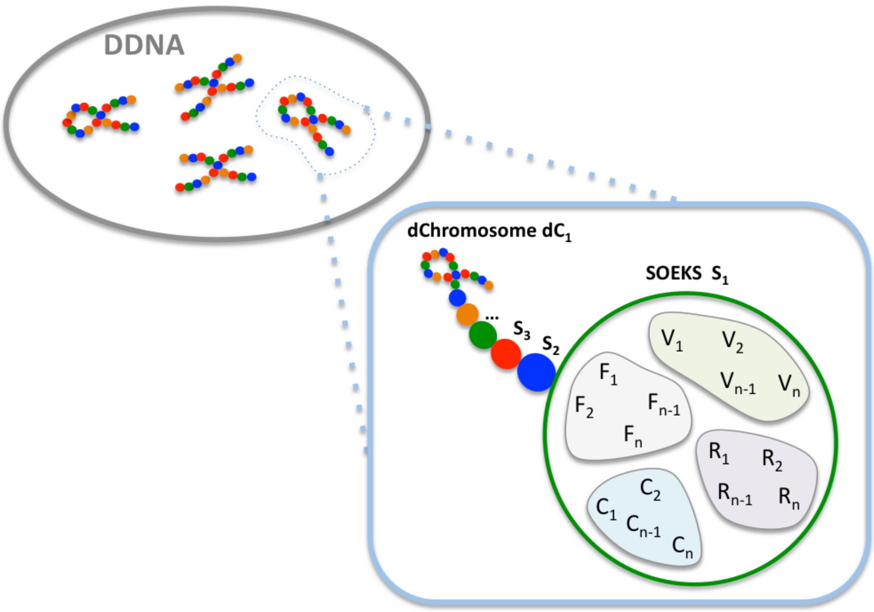

Figure 2.8: Graphical representation of a Decisional DNA

The proposed experience acquisition is inspired in the way DNA stores and transmits information and knowledge. In nature, DNA contains "...the genetic instructions used in the development and functioning of all known living organisms. The main role of DNA molecules is the long-term storage of information. DNA is often compared to a set of blueprints and the DNA segments that carry this genetic information are called genes." [Dru95]. The philosophy of our approach is an architecture to support discovering, adding, storing, improving and sharing information and knowledge among agents, machines, and organisations through experience.

The above motivated a new bio-inspired approach to experience acquisition, modeling and reuse that we propose to apply to enhance clinical decision making processes. Artificial bio-inspired intelligent techniques and systems supporting smart, knowledge-based solutions of real world problems which are currently researched very extensively by research teams around the world, have enormous potential to enhance automation of decision making and problem solving for a number of diverse areas, including clinical diagnosis. Bio-inspired ideas and implementations have a long history and represent successful biomimetic applications [Mar10, Rin07, Ter10]. Nature is full of excellent examples of design and smart organizational/management approaches that produce outstanding results in highly complex situations. The main problem is that most often we simply do not understand how this happens.

Sets of Experience (Decisional Genes) are grouped according to their phenotype (i.e. knowledge category) creating Decisional Chromosomes (dChromosomes), which store decisional "strategies" for a specific category. Therefore, having several SOEKS chromosomes is equivalent to having a complete DDNA strand of an entity[6] containing different inference strategies. The SOEKS and DDNA have been successfully applied in industrial environments, specifically for maintenance purposes, in conjunction with augmented reality (AR) techniques [TSV+07], and in the fields of finances and energy research [SMASC09].

[6] Here an entity can be a company, a decision/design team, or a clinical decision team.

Chapter 3

Clinical Decision Support Systems (CDSS)

This Chapter introduces Clinical Decision Support Systems (CDSS) providing an overview of the current state-of-the art and related technologies for designing, building and maintaining CDSS. The Chapter is structured as follows: Section 3.1 gives basic definitions and background concepts related to CDSS. Finally, Section 3.3 summarizes a historical evolution of CDSS and their architectures.

3.1 Definitions and background concepts

We adhere to the definition of CDSS given in [LWA06] stating that CDSS are active intelligent systems that use patient clinical data to generate case specific advice. According to [Gre11], the main task of CDSS consists of the retrieval of relevant knowledge and patient data (coming from medical devices, evidence provided by the medical community, and clinical guidelines and protocols) and their analysis to perform some action, often the generation of recommendations. The target user can be a physician or any other medical professional, a medical organisation, a patient or patient's caregivers or relatives.

The goals of CDSS are: (i) to facilitate assessment of patient data, (ii) to foster optimal decision making, problem solving and acting, in different contexts and tasks (such as diagnosis and treatment), ensuring that decision makers have all the necessary knowledge to make a correct decision, and (iii) to reduce medical errors [BL07, PT06].

A wide variety of tools can be included in CDSS, some examples are: (i) computerized alerts and reminders, (ii) clinical guidelines, (iii) order sets, (iv) patient data reports and dashboards, (v) documentation templates, (vi) diagnostic support, and (vii) clinical workflow tools [OTM+07]. The technologies in which such tools and interventions are based are sparse (e.g. data mining techniques, communication protocols, knowledge acquisition techniques, semantic representation and reasoning, etc).

In the general domain of Decision Support Systems (DSS), Power [Pow08] categorizes the different approaches in five groups:

(i) model-driven systems, based on quantitative models which are defined by limited data and parameters that decision makers need when analyzing a situation [Mat09]; for instance, a queueing systems model may be used to make decisions on the structure of some information flow process or communication network design.

(ii) data-driven systems, based on the access and manipulation of huge amounts of data [Rin11]; often this kind of DSS needs some kind of data mining techniques, and its use is restricted to the high level management. Decisions based on data mining would affect the whole policy of an institution for a long time period. For instance, the focus on an specific disease to build facilities and resources to treat and/or research about it.

(iii) communications-driven systems, which use network and communications technologies to facilitate decision-relevant collaboration and communication [For10]; these kind of DSS architectures emphasize social interaction to reach a decision, they can be termed also "group DSS" [Blo12]. For instance, in an emergency situation, such as a natural catastrophe, several independent agencies must be performing independent but coordinated decision that need such communication-driven DSS.

(iv) document-driven systems, which use computer storage and processing technologies to provide document retrieval and analysis [Pow08]; these kind of DSS are based on natural language processing techniques for the navigation over a database of documents, which can be research papers, standards, process guidelines, books, etc. For instance, such systems would allow crawling in the medical literature for specific updated documents about a disease or treatment.

(v) knowledge-based systems, that benefit from a symbolic representation of knowledge about a particular domain, and the ability for reasoning about solutions of problems within that domain, [Kal03]. These systems use ontology models and reasoning tools for knowledge representation and problem solving. In this Thesis, we focus our work on the latter typology.

3.2 Knowledge-based CDSS

Knowledge-based CDSS have been broadly reported in the literature. Examples are the Bayesian reasoning for general CDSS Iliad, presented by Warner [War89]; the diagnostic mammography system Mammonet based on bayesian networks presented by Kahn et al [KRW+95]. Amongst other works based on production rules we can mention the IMM/Serve immunological CDSS built by Miller et al [MFS+96]. More recently, [WWK13] presented a web based CDSS that follows a case-based approach, in which editors were provided for knowledge manual maintenance; [ASL11] described an architectural and data model for CDSS, integrated to the clinical system; [CDV08] presented a knowledge-based CDSS for Oncology, where both an ontology and a ruleset were proposed; In [BRH09] an OWL DL ontology for a preoperative risk assessment CDSS was presented. The proposed system was based on a DL reasoner and a rule engine that provided patient preoperative risk assessments.

The general model of Knowledge-based CDSS proposed by Berner et al. [BL07] shown in Figure 3.1, consist of 4 elements: (i) an input, (ii) an output, (iii) a Knowledge Base and (iv) a reasoning engine.

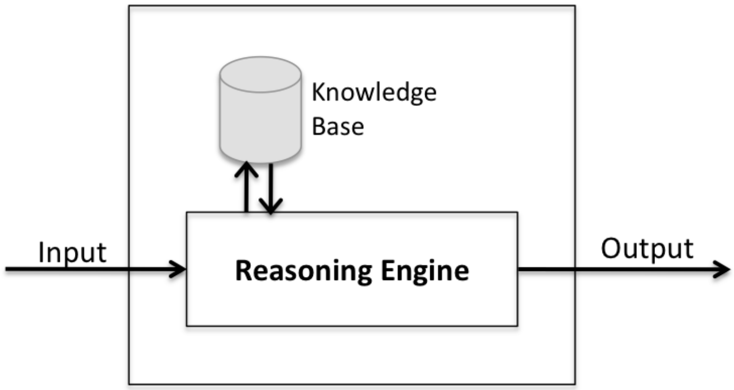

Figure 3.1: Model for knowledge based CDSS

3.2.1 CDSS input

The CDSS input consists of the patient clinical data for which recommendations are requested. Such data are generally specified in a controlled vocabulary, in which the different variables and their possible values are previously stablished.

Patient information can also be provided directly by the Electronic Health Record (EHR), which is a systematic collection of digital health information about individual patients or populations, including demographical data, medical history, medication and allergies records, immunization status, laboratory test results, radiological images, vital signs, personal statistics such as age and weight, and even billing information. There exist several standards for EHR that guarantee the interoperability and comprenhensibility of the data, such as ISO EN 13606, HL7 and openEHR [SBH+06].

3.2.2 CDSS output

The CDSS output is usually provided as a list of possibilities ranked in some order of probability, such as the most likely, the less likely, and the most save or risky. Depending on the application domain and the purpose of the system, the most likely possibility could not be interesting for clinicians, as such could be trivial or inmediate for them. However, clinicians are in general interested in having a broader spectrum of alternatives to consider. Hence, some knowledge-based CDSS are focused on providing less likely options, together with the evidence supporting such recommendations.

3.2.3 Knowledge Base

The Knowledge Base consists of some form of medical knowledge. The representation of such knowledge may be obtained by means of applying different techniques, depending on the technology of the Reasoning Engine. A very common technique is the modeling of the knowledge domain in ontologies (see Chapter 2). This variant contains the description of the different elements included in the domain, their relationships and instances. The codification of the criteria for solving the different decisions and aspects of the domain may also be included. Greenes [Gre11] describes three different methodologies for the generation of a Knowledge Base, depending on domain and criteria to be encoded in the Knowledge Base:

Data based approach Depends largely on the availability of data on which some kind of learning or statistical inference process could be performed in order to obtain similarities between data patterns. The resulting systems classify data into categories or output classes, or perform some kind of functional prediction by regression. The main disadvantage of such approach is that patterns obtained from data of a certain population are not trasferable to other populations.

Meta-analysis It consists of methods focusing on retrieving different studies following inclusion criteria, in order to contrast and combine them for identifying patterns among results. The procedure is very strictly protocolized and evidence can be provided to sustain the identified effects or conclusions with strong agreement in the literature. Evidence Based Medicine is based on this kind of approaches, and it is becoming a popular practice [Str11]. The meta-analysis methodology is out of the scope for us, as it is performed directly by humans and not computer systems.

Human intensive techiques This approach consists on the modeling of the knowledge domain by means of the input provided and consenssed by domain experts. In our work we follow this approach, as it does not require any training of the system, and thus is data independent and transferable to other populations.

3.2.4 Reasoning Engine

The Reasoning Engine combines the input data and the medical knowldge, according to some logical scheme, for generating the output. In Chapter 2 different schemes and technologies have been presented, that suit the core of the Reasoning Engine: statistical approaches, Bayesian Networks, Neural Networks, and production rule systems.

3.3 Historical overview and architectures

As already mentioned, CDSS have been broadly studied in the literature for the last 50 years. Wright et al. [WS08] have reviewed CDSS along those years concluding that the evolution of architectures for CDSS has followed four phases: standalone, integrated, standards-based, and service models. Based on such cathegorization, in the following Sections we provide a brief overview of the most relevant CDSS until the time of writting of this dissertation.

Stand-alone CDSS

Standalone CDSS run separately from any other system, such as clinical information management systems containing the clinical information from patients and cases. Thus, a physician has to intentionally enter the required information and ask for the computerized support, a time consuming process. Usually the system is not proactively supporting decision making. On the bright side, these CDSS are very easy to share [WS08]. Beginning in 1959 [WS08], the main steps in the historic evolution are the following:

- A card sorting system for differential diagnosis [LL59] proposed by Ledley et al in 1959 is widely recognized as the first CDSS.

- Kodlin et al [KC72] proposed a card system (on similar approach to the latter) to be sorted by the patient depending on their answers to questions, enhanced with computer processing for automated diagnosis in multiphasic screening.

- In 1969 appeared the first CDSS to include therapy suggestions, in addition to diagnosis in the work of [Ble69]. This system was oriented to acid-base disorders, requesting and collectiong the required data and producing an evaluation note.

- In 1972, a very effective probabilistic model and computer system for abdominal complaint diagnosing was presented by [dDLS+72].This approach decreased the error rate in half, compared to the performance of senior clinicians. proving inequivocaly a certain degree of maturity in the technology.

- In 1975 MYCIN, the first medical Expert System, was presented by Shortliffe [SDA+75]. This approach provided therapy advice for patients with infections. It contained a ruleset provided by domain experts, based on which recommendations about antibiotic prescription would be provided. Some years later Shortliffe would present ONCOCIN [SSB+81], meant to assist oncologists with chemotherapy management.

- In 1980 Miller et al proposed INTERNIST-I [MPM82], a diagnostic decision support system covering the entire field of internal medicine, being the first approach that was not limited to a single domain.

- In 1983 the ATTENDING system was proposed [Mil83], a critiquing system for anesthetic plan management. The inputs of the system were: (i) patient clinical data, (ii) the planned surgical procedure and (iii) an anesthetic plan outlining the agents and techniques to be used. The result was a critique of the inputted plan, discussing the risks and benefits of the proposed or other reasonable approaches.

- In 1987 DXplain [BCHH87] was presented, with a clear aim to the explaining of the process involved in reasoning.

Integrated CDSS

In order to overcome the main issues arose with stand-alone system, integrated CDSS into clinical systems were proposed. They behave just the opposite.

- The first integrated system was HELP, in 1967 [KGP12], which added to medical record a monitoring system able to generate automatically alarms and reports. Since 1967 HELP has been further developed, and currently is in use in most Intermountain Health Care's Hospitals.
- The COMPAS system was presented in 1989 by Sitting [SGP+89], for ventilator management.
- In 1990 Gardner et al presented a system for blood product ordering [GGL+90].
- In 1998 Evans presented an antibiotic advising system [EPC+98].
- In 1973 RMRS system was presented [MMJ+77], which provided suggestions based on a large ruleset. It was tested in relevant clinical trials.
- The WizOrder system presented in 1998 [GM98] is in use at Vanderbilt, and McKesson commercializes it as Horizon Expert Orders
- The Brigham Integrated Computing System (BICS) [TSF+93], in use at the Brigham & Women's Hospital, includes pathway support (e.g. data entry and ordering tasks).
- The Computerized Patient Record System (CPRS) [BTL+99] of the Veterans Health Administration has also been a significant advance of the state of the art.

Standard-based CDSS

Standards-based systems aim to the standardization of the computerized representation, encoding, storing and sharing of clinical knowledge and decision support content [WS08]. There are several standards offering a different focus:

- Arden Syntax [Zho08], firstly developed at 1989, standardized clinical decision support content.

- the Guideline Interchange Format (GLIF) [OMGM+98] focuses on complex multi-part guidelines, including complex clinical pathways that take place in phases or over time. In [WPT+04]a general-purpose execution engine for executing GLIF guidelines was presented, but has not yet been implemented in any commercially available system.

Service-based CDSS

Service models separate clinical information systems and CDSS [WS08], and integrate them, while still using standardized service-based interfaces. The standard interface can be either located in front of the clinical system, so that any decision support system that understands the standard can infer (i.e. HL7 vMR [JTMP01]), or located in front of the decision support system, as a service for third party clinical systems that understand this standard and can ask for aid (i.e. HL7 DSS [Kaw07]).

- In 2004 the Shareable Active Guideline Environment project (SAGE) [RBT+04] was presented, which places an API in front of the clinical system. In this way, a SAGE rule would interact with any clinical system that supported the SAGE-compliant API. This approach of placing a standardized interface in front of the clinical system is called a Virtual Medical Record (VMR) [JTMP01].

- In 2005 SEBASTIAN [KL05] was presented, placing a standardized interface in front of clinical decision support modules. In this approach, any clinical system understanding the SEBASTIAN protocol could make queries of centralized Decision Support Services. SEBASTIAN modules are located on the Internet, and thus they can be shared by more than one hospital, allowing for greater efficiency. SEBASTIAN has evolved through HL7 [LKA+07] as the HL7 DSS, which uses the HL7 Version 3 Reference Information Model (RIM) as its patient data model (resolving vocabulary challenges of SEBASTIAN.

3.4 Challengenges of current CDSS

As mentioned before in this Chapter, since the 1970s numerous CDSS and technologies have been proposed [Blo12, Hol08, WS08, BL07], but the integration of such systems in daily clinical environments has not been fully achieved yet [OTM+07]. Factors driving this lack of success have been studied in the literature [PT06, SWO+08, KHBL05, OTM+07, DE10, HXB03, Gre06, LWA06]. With the following paragraphs we would like to summarize the main challenges that current CDSS need to bridge into the following:

CHALLENGE 1: Computerization of clinical decision support

Historically, decision support recommendations have been formalized, shared and reused by the medical community (*i*) in medical books and journals, (*ii*) in specialized conferences, (*iii*) in clinical protocols that determine the disease-specific recommended guidelines, and (*iv*) in meetings were the characteristics of a disease or the interventions to follow for different cases have been discussed. Recommendations shared by these means are correct and have a strong medical validity, but they are not usable in day-to-day decisions due to the retrieving effort required.

Kawamoto et *Al.* suggested that decision support should be computerized and not paper-based [KHBL05]. Current efforts in computerization of decision support are mainly focused on the development of computer-based medical guidelines and protocols [IA]. However, actual knowledge representation models for clinical guidelines do not prioritize reasoning as it could be argued that they are mainly focused on alignment and integration of data. These approaches, although signifying a stepping stone towards the inclusion of semantics in CDSS, still lack of the exploitation of the knowledge embedded in the aligned data, and thus, improvements in knowledge representation and reasoning capabilities are still in need. In addition, current CDSS do not deal with the extraction of medical experience from day-to-day decision making, which is reflected in the clinical history of patients.

CHALLENGE 2: Clinical workflow integration

Decision support should be integrated into the clinical workflow [HXB03, KHBL05, DE10, BKW+03]. The importance of workflow integration is evident in the direct impact in the minimization of time consumption during the introduction of patient data and results. In this sense, efforts should be done for the integration of CDSS with clinical systems already present in hospitals and medical centers [HXB03, DE10]. Additionally, CDSS should be presented as complete solutions that assist clinicians during all different tasks of their daily duties, and not only during specific activities. This fact would promote and normalize the use of clinical decision support.

CHALLENGE 3: Maintainability and extensibility of CDSS

CDSS should be easy to maintain and extend, when new knowledge is fed to the system [PT06]. Cost-saving solutions are needed for the updating process of the underlying knowledge model. For that purpose, there is an urgent need in creating knowledge representations that are sufficiently transparent to be understood directly by domain experts [Gre06]. At the same manner, easy-to-use, and technology-transparent tools for domain experts need to be developed. Ease of use is not only conveyed towards GUI enhancements, but also in allowing medical practitioners to visualize and edit the knowledge models and criteria for the reasoning in a simple, yet powerful way.

Apart from that, the knowledge and criteria embedded in CDSS should evolve with daily experiences [BL07]. A reason for the aforementioned fact is medical training, which is based on the concept of experience-based learning: graduate doctors expend a 4-5 year-long internship before they became fully qualified doctors, time period in which they learn to work under the supervision of a team of experienced doctors. Following this same paradigm we identify the need of having systems capable of acquiring day-to-day experience and learning from that in order to improve the knowledge and criteria of the CDSS.

Since physicians are already using knowledge and their own experience to make decisions, in our work we implicitly show that experience-based systems are yet to be considered as a new category in [Pow08]. With that challenge in mind, we propose in our work the use of experience modeling and reasoning techniques, such as SOEKS and DDNA [CS09, SMASC09]. With the use of such technologies CDSS would not only integrate in the decision making process of physicians during clinical workflow, but they would also able to learn from the everyday experiences mimicking physicians learning at procedural level.

CHALLENGE 4: Timely advice

Clinical decision support should be provided at the place and time when it is needed [HXB03, PT06, BKW+03]. The aforesaid leads to the need of fast reasoning processes, aimed to provide real time, or quasi-real-time, responses from those semantically enhanced clinical decision support systems.

CHALLENGE 5: Evaluation of costs and effects of CDSS

Costs and effects of the implementation of CDSS in real clinical environments should be measured and evaluated [PT06]. In this sense, mechanisms for the quantitative and qualitative evaluation of the performance of the system, as well as of the quality of the knowledge and the models in it should be provided [LWA06].

CHALLENGE 6: CDSS modules and services architecture

An architecture that allows the sharing and reusing CDSS modules and services [SWO+08] is needed. In our work, we propose an architecture based on semantic technologies, which provide the needed expressivity, modularity and reasoning capabilities for building such system.

Part II

Methodological contributions

Chapter 4

Reasoning and recommendation generation

In order to propose recommendations, we use a resoning over domain approach, in the context of this Thesis. Our aim is to gather and infer conclusions from production rules. In this Chapter we propose our reasoning system and the generation of decisional recommendations. In order to rationalize our approach we present a specification that will sustain the logic models supported in the Knowledge Bases we use for persistence. We introduce first the underlying knowledge model and then the necessary extensions that will convey towards the reported needs solution. The starting point of our approach is the work of Toro et al. [Tor08] on Reflexive Ontologies (RO). We extend RO by including the handling and reasoning that production rules provide. The reasoning process is detailed in a specification, that allows us to introduce and discuss in deep implementation aspects already achieved in the course of the realization of the research projects that supported this Thesis realization.

This Chapter is structured as follows: Section 4.2 introduces the specification of the underlying knowledge model from which recommendations are inferred. Section 4.3 details the process of generation of decision recommendations. Section 4.4 studies implementation details. Finally, Section 4.5 discusses some relevant aspects of our approach.

4.1 Motivation and generalities

Decision Support Systems (DSS) are software tools aimed at aiding decision makers, by providing different mechanisms to help them during decision making tasks. DSS cover a wide span of tools based on several technologies and approaches. Particularly, [Pow08] identifies knowledge-based DSS (k-DSS) as tools with specialized problem-solving expertise which allows them to provide decision recommendations to users.

Decision recommendations are a set of alternative options for actions or diagnosis that have been calculated by the system according to previously stablished criteria. Recommendations are ranked and presented to system users, so that they can easily analyze the different suggested choices, as well as their proofs. In this sense k-DSS act as black boxes that output decision recommendations for a given input data set.

Reflexive Ontologies defines an abstract knowledge structure (i.e. an ontology and its instances) endowed with the capacity of maintaining an updated image of every query performed on it [Tor08]. That is, the RO maintains the history of queries and the actual collection of instances that answer each query. Hence, a virtual deamon is associated to each query. This deamon evaluates the potential of changes on the query answer introduced by any operation on the data layer of the systems. The purported advantage of RO is that of speeding up query response. It also implies that some knowledge generation can be produced, i.e. new rules can be generateed, on the basis of query interaction. This potential behavior was termed "autopoietic" in the original proposal [Tor08], following the biological inspiration of Maturana in his seminal work [Mat80]. Though it has not been developed in its full extension, in this Chapter we will work towards the formal specification of both the RO with autopoietic behavior, which we call Extended Reflexive Ontologies, and the reasoning system that would support them. Other interesting properties of RO are the ability to maintaing the integrity of the collection of queries and self-creation. We borrow from [Tor08] Figure 4.1, which shows the logical structure of a RO, which is, basically, a conventional ontology extended with a reflexive structure (mainly composed by the query instances in the left part of the image). As can be seen, every query (Q_p) is related to at least one class of the ontology (C_i) and one

-or more- instance (I_k).

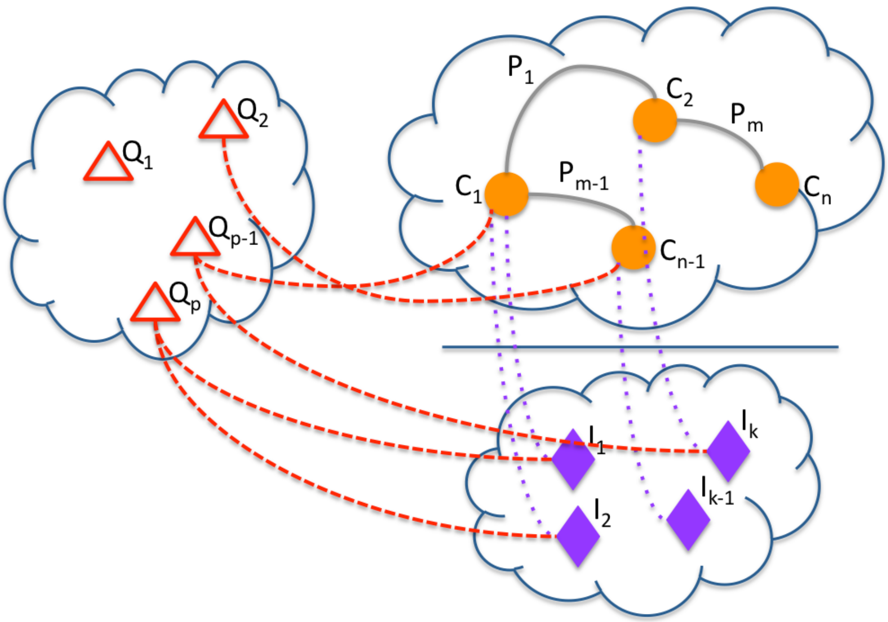

Figure 4.1: Schematic representation of the structure of a RO

4.2 Knowledge model specification

This section presents a specification of the Knowledge Model, denoted $\aleph$, in each of the cases, i.e. baseline Ontologies, Reflexive Ontologies and Extended Reflexive Ontologies (ROX). This specification will serve to give a fomal specification of the Reasoning process $\Re$ in the next Section. The goal is to obtain a high level but concrete specification of the semantic constructs.

4.2.1 Domain Ontology

Let $O = \langle C, P, I \rangle$ denote a domain ontology, whose elements are a set of classes $C = \{C_1, C_2, ..., C_{N-1}, C_N\}$, a set of properties $P = \{P_1, P_2, ..., P_{N-1}, P_N\}$, and a set of instances $I = \{I_1, I_2, ..., I_{N-1}, I_N\}$.

- A class C_i defines a group of individuals that share common properties. Classes in C can be hierarchically organized.

- A property P_i defines relationships either (i) between sets of individuals, or (ii) from set of individuals to data types. When P_i relates instances of two different classes or instances of the same class it is called an Object Property, P_i^o. Likewise, when P_i relates instances of a class to instances of data types (e.g. Integer, String, Float), it is called a Datatype Property, P_i^d. A property P_i can be formalized as a map $P_i : \mathcal{D}_i \to \mathcal{R}_i$
  - The property domain $\mathcal{D}_i$ specifies the individuals to which it can be applied. For instance, if the domain of a property P_i is a class C_i the instances to which it is applied must belong to C_i.
  - The property range $\mathcal{R}_i$, specifies the individuals that could be assigned as values.
- An individual I_i defines an instance of a class C_i, we use the notation $I_i \in C_i$ to specify this instantiation. Properties P_i between classes are mapped homomorphically to properties relating individuals.

Example Figure 4.2 depicts an example ontology where $C = \{C_1, C_2, C_3\}$, $P = \{P_1^o, P_2^d\}$, such that $P_1^o : \{C_1\} \to \{C_2\}$ and $P_2^d : \{C_3\} \to$ Integer, and $I = \{I_1^{C_1}, I_2^{C_1}, I_3^{C_1}, I_4^{C_2}, I_5^{C_2}, I_6^{C_3}, I_7^{C_3}\}$. We have drawn a virtual division separating the semantic level from the data level. Notice also that we have drawn in the data level the property arrows induced by the properties defined at the sementic level.

4.2.2 Querying the ontology

Individuals of an ontology are instances of the classes in the knowledge structure, so that searching in the space of instances is enhanced by the possibility of reasoning at the semantic level of classes and properties. Hence, an ontology provides semantic enrichment of the data. A query is a search within the ontology that returns a collection of instances satisfying a set of clauses.

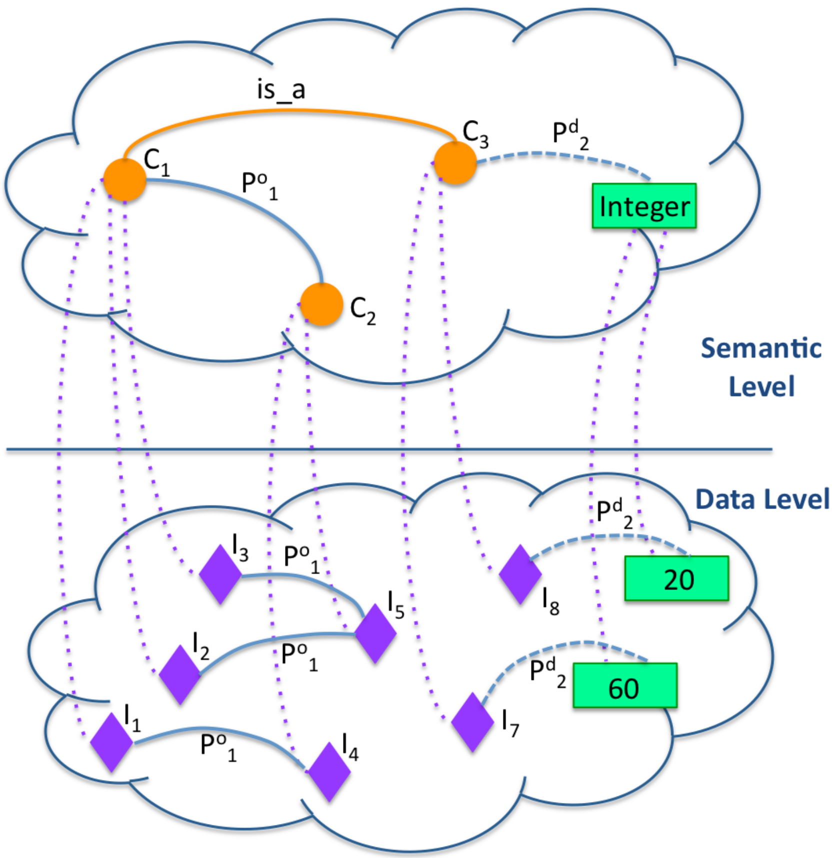

Figure 4.2: Example ontology

We specify these ideas as follows: A query is a pair $Q_i = (q_i, I_{Q_i})$ where q_i are the clauses specifying the chatacteristics of the search, and $I_{Q_i} \subset I$ is the subset of the ontology individuals matching the query clauses. In fact, a query is a map of the form:

$$Q_i : I \xrightarrow{q_i} \mathcal{P}(I),$$

where $\mathcal{P}(I)$ is the power set of I, the set of subsets. We may express the same idea by defining a map σ between q_i and I_{Q_i}:

$$\sigma(q_i) = I_{Q_i}.$$

This notation is more handy in some parts of our description. It can be formulated in terms of the individuals as follows:

$$\sigma\left(q_i\right) = I_{Q_i} = \left\{I_k \in I \left| M\left(q_i, I_k\right)\right.\right\},$$

where $M\left(q_i, I_k\right)$ is a very general predicate that is true when query clause q_i is satisfied by the assignment of values to variables in an individual I_k. We will also say that I_k matches q_i, a terminology more appropriate of searching techniques.

A query clause q_i can be simple or complex, denoted q_i^s or q_i^c, respectively. A simple query clause is specified by a tuple $q_i^s = \langle V_i, m_i, v_i \rangle$, where V_i is a variable, m_i is the comparison operator (i.e. $>, <, =$) and v_i a value of the range of V_i. A complex query q_i^c is specified by n simple queries, combined by logical operators, θ, (i.e. $\vee$, $\wedge$ and $\neg$) which define the relationships among consecutive simple queries:

$$q_i^c = \{(\theta_n, q_n^s)\}_{\forall n},$$

where θ_n is the n-th logical operator (i.e. $\vee$, $\wedge$ and $\neg$), for consistency we assume that $\theta_0 = \emptyset$.

Example As an example to illustrate the notion of query, four different queries are applied over the ontology showed in Figure 4.2:

1. q_1: [Get individuals of class C_1] $\rightarrow$ $I_{Q_1} = \left\{I_1^{C_1}, I_2^{C_1}, I_3^{C_1}\right\}$
2. q_2: [Get individuals of class C_1 that relate to Individual I_4 through property P_1^o] $\rightarrow$ $I_{Q_2} = \left\{I_2^{C_1}, I_3^{C_1}\right\}$
3. q_3: [Get individuals of class C_3 that have an integer value lesser than 20 through property P_2^d] $\rightarrow$ $I_{Q_3} = \left\{I_6^{C_3}\right\}$

Figure 4.3 shows individuals matching such queries.

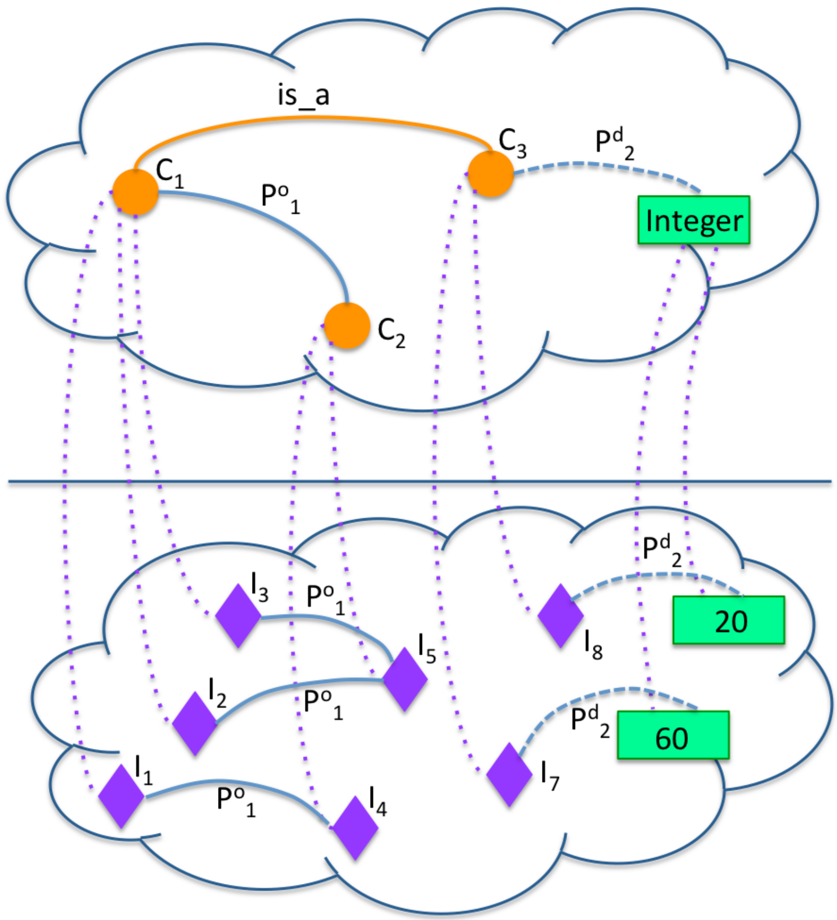

Figure 4.3: Example ontology showing the subsets of individuals matching each query in the query.

4.2.3 Rules

The atomic knowledge encoding is the *rule*, which states the consequences of the search performed on the semantically enhanced data. Rule consequentes are actions involving variable value assignment or recommendations. A rule r_k is composed of a query clause and the consequent actions. Each rule is formalized as a tuple $r_k = \langle A_k, S_k, L_k, W_k, B_k \rangle$, where

(i) A_k is the set of conditional clauses (antecedents), that are equivalent to the q_i part of the queries,

(ii) S_k is the set of actions corresponding to the **THEN** consequents,

(iii) L_k is the set of actions conrresponding to the **ELSE** consequents,

(iv) W_k is the rule salience (aka weight), defined as a real number $W_K \in [0, 1]$, and

(v) B_k is a generic notation for for application-dependent ancillary information that can be associated to rule.

A special kind of action is the assignement of a value to a variable, i.e. $V_l = v_l$. In the context of a rule, this action is restricted to individual instance fulfilling the antecedent clause of the rule, for the **THEN** consequent, or its negation, for the **ELSE** consequent. We assume that the aforegoing assignement expression is equivalent to $I_k.V_l = v_l$, wehre the dot notation specifies the fact that the variable is an attribute of the individual instance, which may fulfill the antecedent clause or not, as discussed before.

Figure 4.4 depicts the structure of a rule.

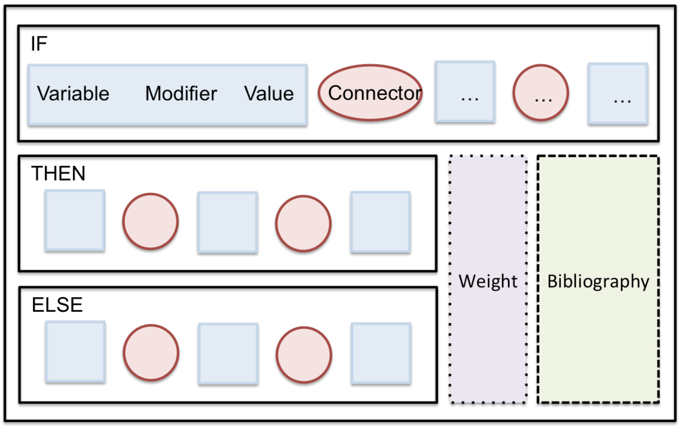

Figure 4.4: Rule syntax

To identify each of the different types of decisions (recommendations) that can be produced by the search and reasoning over the sematically enhanced data we introduce the Decision Domain, denoted d_i. Each d_i is associated to a property P_i in the ontology, we denote this association as follows: $d_i \hookrightarrow P_i$, because it is not strictly a map. We say that a rule r_k is oriented towards a Decision Domain d_i, when the THEN and ELSE consequents S_k and L_k, respectively, refer to the Property P_i associated with d_i. Each rule r_k is oriented towards some d_i and both consequents, S_k and L_k, must refer to the same set of d_i.

4.2.4 Reflexive Ontologies

Reflexive Ontologies (RO) proposed in [Tor08] involve the capability of an ontology based knowledge management system of maintaining a history of the queries performed on it, and to derive new knowledge (i.e. new rules) from the history of the user interactions. The original proposal has been subject to evolution through a series of implementations related to specific applications. In this Section, we present the first actual attempt to provide a formal specification, which, though remaining abstract, is concrete enough to discuss the consequences and degree of implementation. We will not discuss computational speed up issues, which are tackled in Chapter 8 in a very specific application environment.

Formal Specification of Reflexive Ontologies

A reflexive ontology RO is a tuple $RO = \langle O, Q^t \rangle$, where O is a domain ontology and $Q^t = \{Q_1, Q_2, ..., Q_{N-1}, Q_N\}$ is the set of queries that have been performed over the set of instances, I, and classes, C of the ontology up to time t. Therefore, the RO is a time varying structure in two senses:

1. Its query set Q^t will be growing in time: Each new query will be added to it.
2. Changes in the instance layer, i.e. by the edition of an individual, will be reflected on the queries that include it.

The properties of RO are specified as follows.

Query retrieval: The RO must be able to detect and store every new query -and subquery- performed on it. Let us denote Q_{i^*} a new query posted by the user.

$$\neg\exists q_i \in Q^t \text{ s.t. } q_i = q_{i^*} \Longrightarrow Q^{t+1} = Q^t \cup \{Q_i\}\,.$$

On the other hand, if the query has been already posted and answered, an updated anwer will be provided

$$\exists q_i \in Q^t \text{ s.t. } q_i = q_{i^*} \Longrightarrow I_{Q_{i^*}} = I_{Q_i}.$$

Integrity update: The system must be able to actualize the query set every time a new individual is added to, removed from or modified within the ontology. Let us denote I_k^t the variable value assignment of the k-th data instance at time t. The integrity update means that, at any time, if an instance satisfies the clause of a query, then it belongs to the data associated to the query:

$$\forall q_i \in Q^t \text{ s.t. } M\left(q_i, I_k^t\right) \Longrightarrow I_k^t \in I_{Q_i}^t$$

This specification is purely declarative. If we want to advance something on the mechanism that may implement such property, we can state what happens for each change in instance layer. For the sake of notation, let us assume that the introduction of a new instance at time t can be formalized as $I_k^{t-1} = \emptyset$ and $I_k^t \neq \emptyset$. Also, the following holds always $M(\emptyset, q_i) = F$. Hence, the integrity update can be specified as follows:

$$I_k^{t-1} \neq I_k^t \Longrightarrow \begin{cases} \neg M\left(q_i, I_k^{t-1}\right) \wedge M\left(q_i, I_k^t\right) & I_{Q_i}^t = I_{Q_i}^{t-1} \cup \left\{I_k^t\right\} \\ M\left(q_i, I_k^{t-1}\right) \wedge M\left(q_i, I_k^t\right) & I_{Q_i}^t = I_{Q_i}^{t-1} - \left\{I_k^{t-1}\right\} \cup \left\{I_k^t\right\} \\ M\left(q_i, I_k^{t-1}\right) \wedge \neg M\left(q_i, I_k^t\right) & I_{Q_i}^t = I_{Q_i}^{t-1} - \left\{I_k^{t-1}\right\} \end{cases}.$$

The three possibilities specify all possible casuistry. The first case is when the data instance was not included in the past version of the query, but its new values do match the query clause, then the instance is added to the query data. The second case is when the data instance was already in the query, but it has changed, then the instance must be updated in the query (i.e. the old version removed and the new one added). Finally, when the instance no longer matches the query clause, then it must be removed from the query data.

Self reasoning over the query set: This property states the ability to perform some kind of query result mining. Some possible ways of self-reasoning are:

i) discover patterns of queries. As an example, assume that some pair of queries Q_{i_1} and Q_{i_2} have a non empty intersection of their corresponding data instances, i.e. $I_{Q_{i_1}} \cap I_{Q_{i_2}} \neq \emptyset$, then we can add a new query corresponding to this intersection $Q_{i^*} = \left(q_{i_1} \wedge q_{i_2}, I_{Q_{i_1}} \cap I_{Q_{i_2}}\right)$.

ii) recommend ontology refinement based on the queries performed over the system. As an example, consider the case when some class is never searched by any query, it may well be denoted obselete or redundant, i.e. if $\forall i, I_{Q_i} \cap C_j = \emptyset$ then we may propose to remove C_j from the ontology.

Remaining properties The support for logical operators is supported in the definition of the rule system, and the autopoietic behavior is a property that is related to the second order reasoning over the ontology and the alignment with third-party tools generating sinominia, equivalent concept matching, statistical and fuzzy analysis.

4.2.5 Extended Reflexive Ontologies

In this Thesis we propose the Extended Reflexive Ontologies (ROX) whose main feature is the maintenance of the rule and recommendation history along with the query history already keep by the RO. Figure 4.5 shows the structure of the Extended Reflexive Ontologies (ROX).

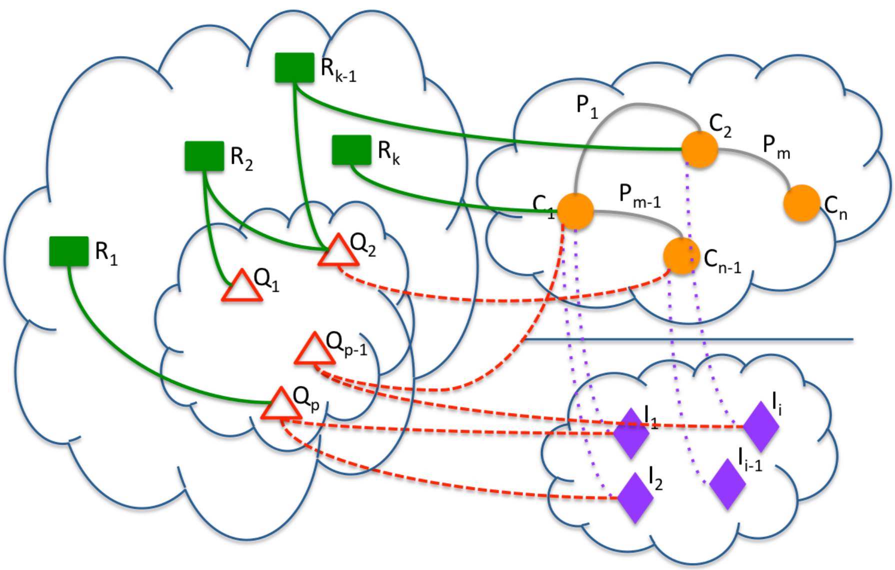

Figure 4.5: Extended Reflexive Ontologies

A ROX is a tuple $ROX = \langle RO, \mathcal{R}^t \rangle$, where RO is a Reflexive Ontology and

$\mathcal{R}^t = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{K-1}, \mathcal{R}_K\}$ the historical set of rules applied at least once to obtain a recommendation.

Let a rule-recommendation $\mathcal{R}_K$ be defined as a tuple $\mathcal{R}_K = \langle r_k, u_k \rangle$, where

(i) r_k is a rule such that $r_k = \langle A_k, S_k, L_k, W_k, B_k \rangle$, where the antecedent A_k is a query, $Q_k = (q_k, I_{Q_k})$, and Consequents S_k and L_k are corresponding actions taken in the THEN and ELSE parts of the rule, and

(ii) u_k are the output domain assignments asociated to r_k, where $u_k = \left\{ \left(I_U^d, V_d, v_d \right) \right\}$ is a collection of domain variable value assignments $V_d - v_d$, where d is the domain indicator, I_U^d is a set of individuals affected by the domain value assignment $I_{k'}.V_d = v_d, \forall I_{k'} \in I_U^d$.

4.3 Reasoning process description

Two different operations are allowed over the Knowledge Model, $\aleph_t$:

(i) the request for decision recommendations, which is performed by a reasoner, $\Re$, and

(ii) the addition or edition of instances, performed by an editor, ε.

Such operations drive different reasoning mechanisms, depending on the Knowledge Model, $\aleph_t$. Among the following Sections we will analyze the mechanisms required for the cases where $\aleph_t$ is an Ontology, a Reflexive Ontology and an Extended Reflexive Ontology.

4.3.1 Reasoning over the ontology

Request for recommendations

When inputted a Request $J = (I_J, D_J)$, where $I_J \subset I$ are a set of individuals for which recommendations are requested, and $D_J \subset D$ are the decision domains of those recommendations, the reasoner $\Re$ outputs a set of recommendations

$K = \{K_{ij}\}$ for a given Ontology, O, and Ruleset, R. The reasoner may provide several recommendations K_{ij} for each request $J_i = (I_i, d_i)$. Figure 4.6 depicts the input/output diagram of the reasoner for the generation of recommendations, when an ontology is used.

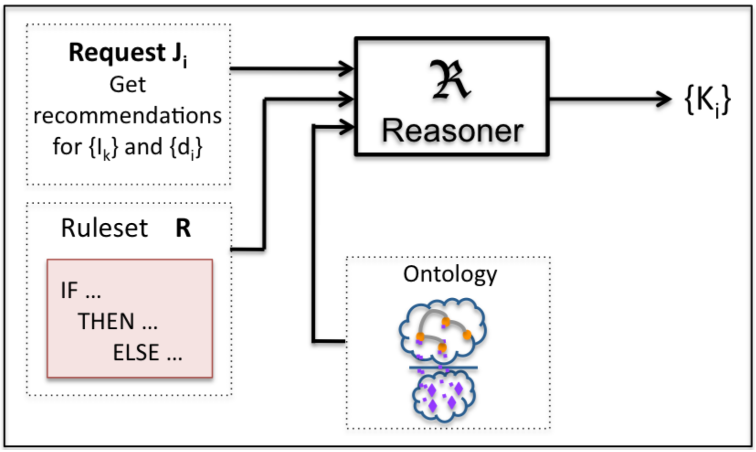

Figure 4.6: Generation of recommendations in Ontology

A recommendation is a tuple $K_{ij} = \langle G_{ij}, W_{ij}, R^{K_{ij}} \rangle$ computed in response to a request couple (I_i, d_i) where:

- The recommendation consequent G_{ij}, which is a collection output domain assignments u_k asociated to rules $r_k \in R^{K_i}$, i.e. $u_k = \{(V_d, v_d)\}$ is a collection of domain variable value assignments, where d is the domain indicator, associated to the consequent of r_k. The value assignment affects the individual instance of the request, i.e. $I_i.V_d = v_d$.

- The weighted probability $W_{G_{ij}} \in [0, 1]$ computed for recommendation G_i,

- A subset of rules $R^{K_{ij}} = \{r_k \,|M\,(A_k, I_i)\}\,; R^{K_{ij}} \subset R$, that provide the supporting evidence for the recommendation consequent G_{ij}.

The output recommendations in K are not ordered, however they are given a different weighted probability $W_{G_{ij}} \in [0, 1]$ computed from the respective weights W_k of the rules endorsing each recommendation G_{ij}. After all recommendations K_{ij} for a couple (I_i, d_i) are calculated, the weighted probabilities $W_{G_{ij}}$ are normalized to guarantee that $\sum_j W_{G_{ij}} = 1$.

Reasoning mechanics Each K_{ij} is built analyzing the sets of instances matching antecedents of rules r_k, whose consequents refer to the same d_i queried in the input request J_i, i.e. $r_k \in R^{K_{ij}}$.

- The matching for each individual I_i and rule r_k is done by translating A_k into a query specification q_i and obtaining the subset of individuals $I_{q_i} \subset I$ that match q_i in ontology O. Let $\overline{I_{q_i}}$ be the set of individuals that do not match q_i in ontology O, such that $I_{q_i} \cup \overline{I_{q_i}} = \mathrm{I}$ and $I_{q_i} \cap \overline{I_{q_i}} = \emptyset$.
- For each individual in I_{q_i} the domain value assignment $V_d = v_d$ in S_k is selected as recommendation consequent G_{ij}.
- On the other hand, for individuals in $\overline{I_{q_i}}$ we select domain value assignment $V_d = v_d$ in L_k.
- Then, W_k is added to $W_{G_{ij}}$ and r_k to $R^{K_{ij}}$.

Addition/Edition of instances in the ontology

The addition of new instances, as well as their edition, is a natural mechanism of the ontology, which can grow up as much as necessary. Let us introduce some temporal notation. Let it be $O^t = (C^t, P^t, I^t)$ the ontology at time. When the instance editor ε adds instance I_i to ontology O^t, returns the updated ontology O^{t+1} where $I^{t+1} = I^t \cup \{I_i\}$. Figure 4.7 illustrates the operation of the editor.

4.3.2 Reasoning over Reflexive Ontologies

Request for recommendations

As stated in the definition above, the Reflexive Ontology stores (i) every query specification q_i executed to the ontology during the recommendations inference process, as well as (ii) the corresponding set of matching individuals I_{Q_i}, into a query pool $Q^t = \left\{Q_1^t, Q_2^t, ..., Q_{N-1}^t, Q_N^t\right\}$, where each $Q_i^t = \left(q_i, I_{Q_i}^t\right)$. The process of reasoning over an RO is illustrated in Figure 4.8.

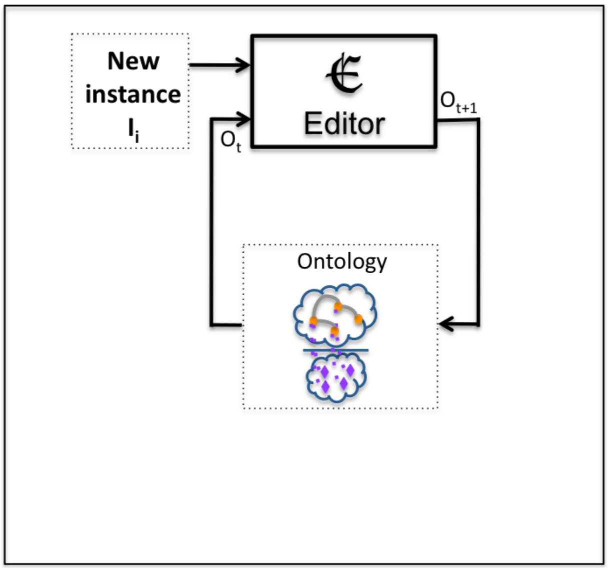

Figure 4.7: Addition of new instance in ontology

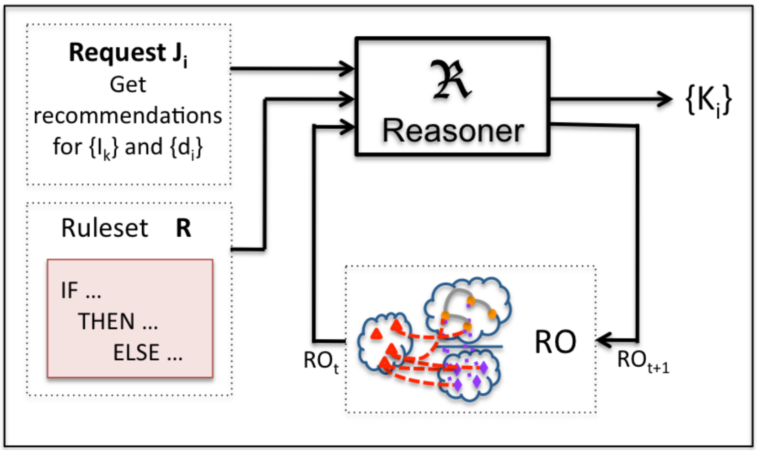

Figure 4.8: Generation of recommendations in Reflexive Ontologies

The basic reconmmendation generation by a reasoner $\mathfrak{R}$ is similar to the process described above for Ontologies: taking as input a Request $J = (I_J, D_J)$, the current Reflexive Ontology, RO^t, and Ruleset, R, the reasoner $\mathfrak{R}$ outputs a set of recommendations $K = \{K_{ij}\}$ for each for each request $J_i = (I_i, d_i)$. A recommendation is a tuple $K_{ij} = \left\langle G_{ij}, W_{ij}, R^{K_{ij}} \right\rangle$ as defined above.

Reasoning mechanics Each K_{ij} is built analyzing the sets of instances matching antecedents of rules r_k, whose consequents refer to the same d_i queried in the input request J_i, i.e. $r_k \in R^{K_{ij}}$.

- The matching for each individual I_i and rule r_k is done by translating A_k into a query specification q_i. We have two cases:

  - The clause q_i corresponds to a query $Q_i \in Q^t$, therefore we have $I_{q_i} \subset I$ updated by the definition of RO. Hence, $\overline{I_{q_i}}$ is automatically given.
  - Otherwisse, we need to search for the subset of individuals $I_{q_i} \subset I$ that match q_i in ontology O. Let $\overline{I_{q_i}}$ be the set of individuals that do not match q_i in ontology O, such that $I_{q_i} \cup \overline{I_{q_i}} = \mathrm{I}$ and $I_{q_i} \cap \overline{I_{q_i}} = \emptyset$. Then, we add the new query: $Q^{t+1} = Q^t \cup \{(q_i, I_{q_i})\}$.

- For each individual in I_{q_i}or $\overline{I_{q_i}}$ the domain value assignment $V_d = v_d$ for recommendation consequent G_{ij} is selected as in the previous Ontology reasoning process.

- Finally, W_k is added to $W_{G_{ij}}$ and r_k to $R^{K_{ij}}$.

Addition/Edition of instances in the RO

Figure 4.7 illustrates the operation of the editor. When the instance editor ε adds instance I_i to the Reflexive Ontology RO^t, we have two operations going on:

1. The updating of the ontology inside the Reflexive Ontology O^{t+1} where $I^{t+1} = I^t \cup \{I_i\}$.

2. The updating of the query collection, which involves the operation of the reasoner $\Re$ to check each $Q_i^t \in Q^t$, so that $I_{Q_i}^{t+1} = I_{Q_i}^t \cup \{I_i\}$ if $M(q_i, I_i)$.

When the editor performs some change in an instance, denote I_i' the edited item, we have similarly two operations going on:

1. The updating of the ontology inside the Reflexive Ontology O^{t+1} where $I^{t+1} = I^t - \{I_i\} \cup \{I_i'\}$.

2. The updating of the query collection, which involves the operation of the reasoner $\Re$ to check the possible situations:

   (a) for each $Q_i^t \in Q^t$, so that $I_i \in I_{Q_i}^t$, $I_{Q_i}^{t+1} = I_{Q_i}^t - \{I_i\}$ if $\neg M(q_i, I_i')$.

(b) for each $Q_i^t \in Q^t$, so that $I_i \in I_{Q_i}^t$, $I_{Q_i}^{t+1} = I_{Q_i}^t - \{I_i\} \cup \{I_i'\}$ if $M(q_i, I_i')$.

(c) for each $Q_i^t \in Q^t$, so that $I_i \notin I_{Q_i}^t$, $I_{Q_i}^{t+1} = I_{Q_i}^t \cup \{I_i'\}$ if $M(q_i, I_i')$.

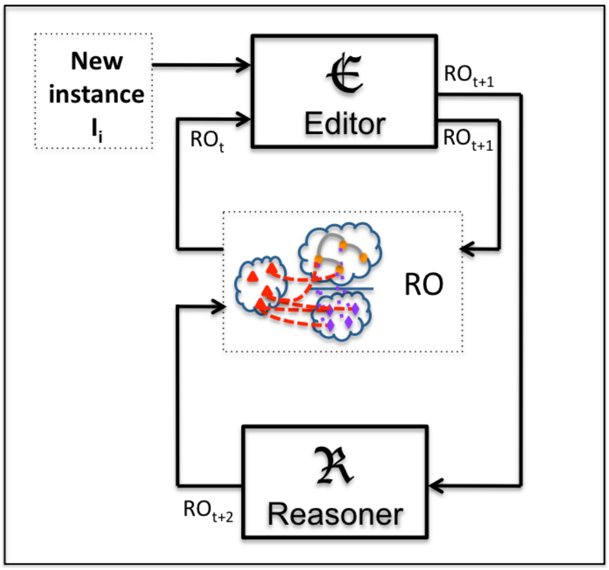

Figure 4.9: Addition of new instance in Reflexive Ontology

4.3.3 Reasoning over Extended Reflexive Ontologies

Request for recommendations

The Extended Reflexive Ontologies approach stores every rule r_k applied to the ontology, into a pool of rules that have been applied $\mathcal{R}^t = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{K-1}, \mathcal{R}_K\}$, such that $\mathcal{R}_k = \langle r_k, u_k \rangle$, $r_k = \langle A_k, S_k, L_k, W_k, B_k \rangle$, and generated domain recommendations $u_k = \left\{ \left(I_{u_k}^d, V_d, v_d \right) \right\}$.

The basic reconmmendation generation by a reasoner $\Re$ is similar to the process described above for Reflexive Ontologies: taking as input a Request $J = (I_J, D_J)$, the current Extended Reflexive Ontology, ROX^t, and Ruleset R, the reasoner $\Re$ outputs a set of recommendations $K = \{K_{ij}\}$ for each for each request $J_i = (I_i, d_i)$. A recommendation is a tuple $K_{ij} = \left\langle G_{ij}, W_{ij}, R^{K_{ij}} \right\rangle$ as defined above. Figure 4.10 illustrates the recommendation generation process by the reasoner.

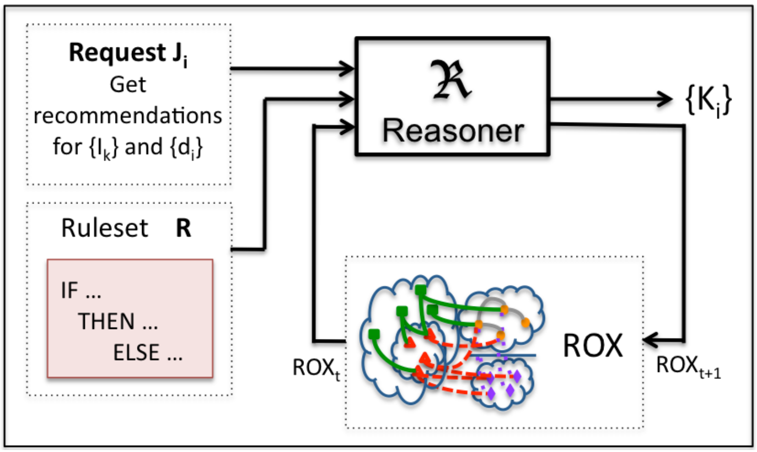

Figure 4.10: Generation of recommendations in Extended Reflexive Ontologies

Reasoning mechanics for recommendation generation K_{ij} is built by analyzing first the stored rules in $\mathcal{R}$. After that the process recalls the reasoning on the remaining rules $R - \mathcal{R}^t$. For each $I_i \in I_J$ we follow the process:

1. For each u_k in $\mathcal{R}^t$ such that $I_i \in I_{u_k}$ we have two situations

   (a) we have a previusly created recommendation K_{ij} such that its recommendation G_{ij} refers to the same v_d as u_k then we add the rule r_k and weight W_k to $R^{K_{ij}}$ and $W_{G_{ij}}$, respectively.

   (b) otherwise we create recommendation K_{ij} such that its recommendation G_{ij} is $V_d = v_d$, the rule set$R^{K_{ij}} = \{r_k\}$, and weight $W_{G_{ij}} = W_k$.

2. For each r_k in $R - \mathcal{R}^t$, if $M(I_i, A_k)$ we compute a new recomendation K_{ij} with $G_{ij} = (V_d, v_d)$ as specificied by the consequent of rule r_k, the rule set $R^{K_{ij}} = \{r_k\}$, and weight $W_{G_{ij}} = W_k$. Besides we update the rule pool of the ROX, as follows, $\mathcal{R}^{t+1} = \mathcal{R}^t \cup \{(r_k, u_k)\}$, with $u_k = \{(I_i, V_d, v_d)\}$.

After computing the recommendations for the given collection $J = (I_J, D_J)$, we may need to perform a compaction process in $\mathcal{R}^{t+1}$ because we may have some redundant u_k which differ only in I_{u_k} which can be compacted into one.

Addition/Edition of instances in ROX

Every time a new instance I_i is added to the ROX by editor ε, the reasoner $\Re$ checks and amy update $\mathcal{R}^t$ as illustrated in figure 4.10.

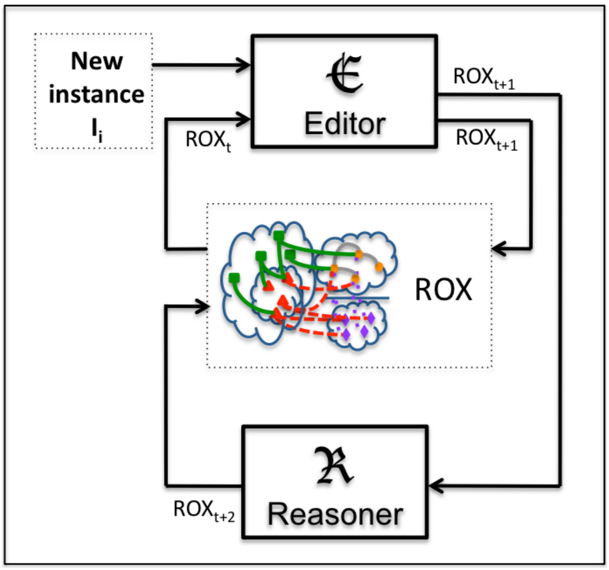

Figure 4.11: Addition of new instance in Extended Reflexive Ontology

When the instance editor ε adds instance I_i to the Extended Reflexive Ontology ROX^t, we have two operations going on:

1. The updating of the ontology inside the Extended Reflexive Ontology O^{t+1} where $I^{t+1} = I^t \cup \{I_i\}$.

2. The updating of the rule collection $\mathcal{R}^t$, which involves the operation of the reasoner $\Re$ to check each $\mathcal{R}^t_k \in \mathcal{R}^t$, so that $I^{t+1}_{u_k} = I^t_{u_k} \cup \{I_i\}$ if $M(A_k, I_i)$.

3. The updating of the query collection, which involves the operation of the reasoner $\Re$ to check each $Q^t_i \in Q^t$, so that $I^{t+1}_{Q_i} = I^t_{Q_i} \cup \{I_i\}$ if $M(q_i, I_i)$.

When the editor performs some change in an instance, denote I'_i the edited item, we have similarly two operations going on:

1. The updating of the ontology inside the Extended Reflexive Ontology O^{t+1} where $I^{t+1} = I^t - \{I_i\} \cup \{I'_i\}$.

2. The updating of the query collection, which involves the operation of the reasoner $\Re$ to check the possible situations:

(a) for each $\mathcal{R}_k^t \in \mathcal{R}^t$, such that $I_i \in I_{u_k}^t$, $I_{u_k}^{t+1} = I_{u_k}^t - \{I_i\}$ if $\neg M(A_k, I_i')$.

(b) for each $\mathcal{R}_k^t \in \mathcal{R}^t$, such that $I_i \in I_{u_k}^t$, $I_{u_k}^{t+1} = I_{u_k}^t - \{I_i\} \cup \{I_i'\}$ if $M(A_k, I_i')$.

(c) for each $\mathcal{R}_k^t \in \mathcal{R}^t$, suc that $I_i \notin I_{u_k}^t$, $I_{u_k}^{t+1} = I_{u_k}^t \cup \{I_i'\}$ if $M(A_k, I_i')$.

4.4 Implementation

Figure 4.12 depicts the classes and properties that are needed in order to extend an ontology O as a ROX.

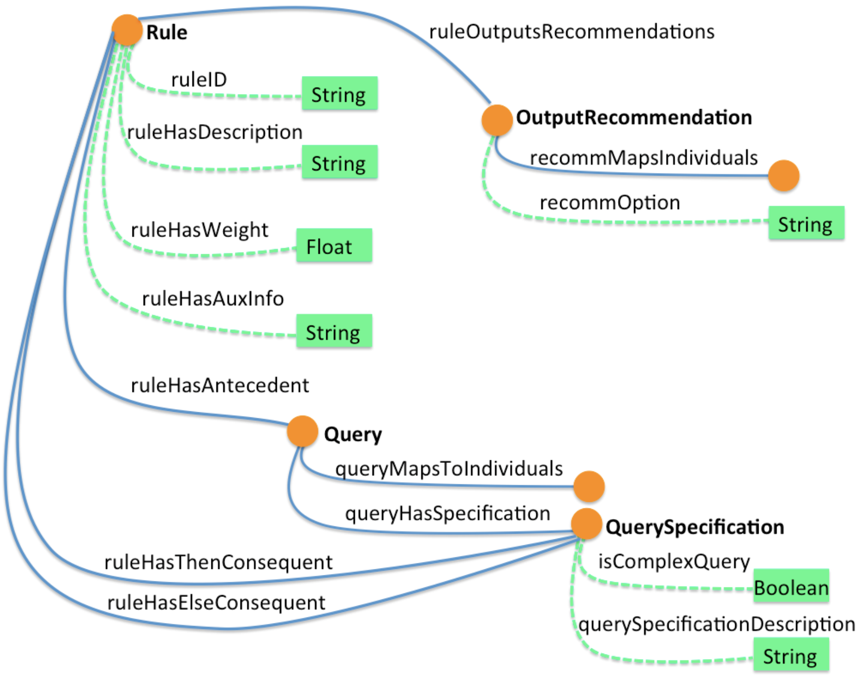

Figure 4.12: Implementation details of Extended Reflexive Ontologies

Classes

- Class **Rule** is the object storing each rule R_i applied.
- Class **Query** is the object storing each query Q_i perfomed to the base ontology.
- Class **QuerySpecification** is the object storing each query specification q_i.
- Class **OutputRecommendation** is the object storing a couple individuals-values, (I^u, v).

Object type properties

- Property *ruleOutputsRecommendations* relates **Rule** instances with instances of **OutputRecommendation**. It is an inverse functional property.
- Property *ruleHasAntecedent* relates **Rule** instances with **Query** instances. It is a functional property, as each rule contains a unique antecedent.
- Property *ruleHasThenConsequent* relates **Rule** instances with **QuerySpecification** instances. It is a functional property, as each rule contains a unique rule then-type consequent.
- Property *ruleHasElseConsequent* relates **Rule** instances with **QuerySpecification** instances. It is a functional property, as each rule contains a unique rule else-type consequent.
- Property *recommendationMapsToIndividuals* relates instances of **OutputRecommendation** with instances of classes of the base ontology.
- Property *queryMapsToIndividuals* relates **Query** instances with instances of classes of the base ontology.
- Property *queryHasSpecification* relates **Query** instances with **QuerySpecification** instances. It is a functional property, as each query has a unique specification.

Datatype properties

- Property *ruleID* relates **Rule** instances with String data values. It is a functional property, as each rule has a unique ID.
- Property *ruleHasDescription* relates **Rule** instances with String data values. It is a functional property, as each rule has a unique description.
- Property *ruleHasWeight* relates **Rule** instances with Float data values. It is a functional property, as each rule has a unique weight.
- Property *ruleHasAuxInfo* relates **Rule** instances with String data values.

- Property *recommendationOption* relates **OutputRecommendation** instances with String data values. It is a functional property, as each option has a unique recommended value.

- Property *isComplexQuery* relates **QuerySpecification** instances with Boolean data values. It is a functional property.

- Property *querySpecificationDescription* relates **QuerySpecification** instances with String data values. It is a functional property, as each query has a unique specification.

Implementation of such extension is done applying the Protégé-OWL API[1], which is based on the JENA Ontology API[2]. First we open the uri corresponding to the ontology that we want to extend and we retrieve the OWL model. We then create each of the four classes above (i.e. **Rule**, **Query**, **QuerySpecification**, and **OutputRecommendation**). Following, the object type properties (i.e. *ruleOutputsRecommendations, ruleHasAntecedent, ruleHasThenConsequent, ruleHasElseConsequent, recommendationMapsToIndividuals, queryMapsToIndividuals,* and *queryHasSpecification*) and the data type properties (i.e. *ruleID*, *ruleHasDescription*, *ruleHasWeight*, *ruleHasAuxInfo*, *recommendationOption*, *isComplexQuery* and *querySpecificationDescription*) described above are created. Then, the generated ontology is saved. Finally, we can feed the Extended Reflexive Ontology with instances.

4.5 Discussion

Advantages of using ROX The enhancement of an ontology in providing self-contained rules and recommendations, relies in the following aspects:

- Speed up of the process of recommendation generation. Each rule r_k, as well as the recommendations u_k provided to each decisional domain d by r_k, are both stored in the Extended Reflexive Ontology (ROX). Thus, when applying

[1] Protégé-OWL API: `goo.gl/NmGirH`
[2] Jena Home Page: `goo.gl/Zru2Ct`

a rule that is already contained in ROX, recommendations do not need to be recalculated. They are only calculated in the case where the rule has never been applied before and are then added to the rule reflexity class.

- Incremental nature of ROX. From the analysis of the previously applied rules and the corresponding attached actions, new rules could be discovered and added to ROX. In this Thesis, such analysis is performed by experience-mining processes executed over a history of stored decisional events. We introduce such processes in Chapter 5.

Application of ROX in Clinical Decision Support Systems In the context of this Thesis the application of ROX in Clinical Decision Support Systems (CDSS) provides a considerable speed up of the process of generation of decision recommendations. Particularly, during patient-recommendations generation many rules are applied to the underlying knowledge bases of the CDSS. As rules tend to be the same for every patient, each time a new patient data is introduced in ROX, the applying rules and recommendations are automatically calculated by the reasoner $\Re$. Thus, when requesting for recommendations, they will be readily available.

Chapter 5

Experience-based learning

Chapter 4 details the reasoning process for the generation of decision recommendations that has been followed in the context of this Thesis. Such process is based on rules provided by domain experts, system users or external rule sources, which feed the reasoner with the required criteria for inferring recommendations that apply for each input data. The quality of the provided recommendations depends largely on the quality of such rules, and thus, rule maintenance and updating become critical when implemented on real-world environments. The difficulties related to maintainance and updating of the ruleset have motivated our work on a methodology for the evolution of production rules. We base our approach on experiential learning and propose reusing experiential facts gained during previous decisions for ruleset evolution. In our work we make extensive use of the SOEKS/DDNA technologies for modeling Decisional Events and in this Chapter we present a formal specification of experience-based learning.

The Chapter is structured as follows: Section 5.1 presents our motivation. Section 5.2 introduces the specification of our experience-based learning model. Section 5.3 presents our experience acquisition and consolidation processes. Section 5.4 proposes an algorithm for the evolution of rule weights based on previous experience. Section 5.5 proposes an algorithm for the fine tuning of rules based on previous experience. Section 5.6 proposed a new rule generation algorithm based on previous experiece. Section 5.7 introduces the extension of the experience model that supports decision traceability. Finally, Section 5.8 discusses quantitative and qualitative evaluation of the presented algorithms.

5.1 Motivation and generalities

The generation of rules covering all possible cause-effects in a specific domain is a very complex task. From the perspective of general systems theory the aforesaid task is unachivable as the perception of the domain causes a perturbation on the domain itself and the observer. The greatest dificulties arise from the observers knowledge about the specifics of the domain, his previous experiences and his ability for abstract modeling [Ber68].

In general terms, specification of a domain is usually made by a group of domain experts, who share their knowledge and reach an agreement [TGP+09]. Not only domain specification is a complex task, but also, to generate facts and rules (statements and antecedent-consequent expressions) is considered a non consistent provider for logical assertions. Rules representing those logical assertions, by nature must be validated, having in fact contradictions and non-simplified statements that would lead to inconsistency.For the aforesaid reasons, it is a reasonable necessity that (i) relevant-to-domain rulesets are as extensible as possible, and (ii) the existence of tools for rule handling is required to facilitate the generation of the rest of non-initially considered cause-effect clauses. The discovery of new knowledge, becomes an urgent line of work when decisional support systems are on duty. Additionally, each rule has a different weight or importance in a decision (i.e. in diagnosing flu, having fieber is more decisive than having cough and mucosity). When the process of rules generation is performed by hand, rule weighting becomes subjective. Objective metrics that could lead in the future to rule comparison are hence requiered for standardization. After having generated the ruleset, it needs to be continously updated to keep up-to-date with scientific advances. Depending on the domain, and the frequency of new discoveries, mainteinance of the rule systems becomes a tedious and very costly task, as it requires continous update by domain experts.

5.2 Specification of the experience data structure

Decisional experience is held in our approach, by means of acquiring a historic of Decisional Events that toke place. As mentioned earlier, in order to model formal Decisional Events we use the SOEKS and DDNA technologies presented in Chapter 2, as already mentioned. SOEKS, is a flexible, independent, and standard knowledge structure, not only for capturing and storing formal decision events as experience, but can also to be used for supporting decision-making and standard knowledge sharing [PW12].

Elements that conform a Decisional Event are captured into a SOEKS object every time a decision is made. In order to specify this, let a SOEKS be a tuple $\mathbb{S}_{\mathrm{t}} = \langle \{\mathcal{V}_n\}, \{C_n\}, \{\mathcal{R}_k\}, \{\mathcal{F}_p\} \rangle$, where

(i) $\{\mathcal{V}_n\}$ is a set of variables involved during the Decisional Event of the instance$I_i \in I$ in the knowledge model (see Chapter 4) to which the decision is oriented, such that v_n are the data values related to I_i by datatype properties P_i^d of the ontology O I_i . Variables formally describe experience-based knowledge structure using an attribute-value language [CS07, PW12]. This is a well-established measure from the foundation of knowledge representation and is the starting point for SOEKS development and composition. Let $\{\mathcal{V}_n\}$ be the set of variables of a domain, where a variable $\mathcal{V}_n = \langle V_n, v_n \rangle$ is composed by a variable specification V_n and a value v_n. Let a variable specification be the tuple $V_n = \langle t_V, \{\mathcal{C}_m\} \rangle$, where t_V is the type of variable (i.e. Integer, Float, Double, String) and $\{\mathcal{C}_m\}$ is a set of constraints, as defined below.

(ii) $\{\mathcal{C}_n\}$ is a set of constraints selecting a subspace ϕ_n of the value range of variables $\mathcal{V}_n$. Constrainsts describe relationships among variables, restricting the possibilities of feasible values. Each constraint is specified as a predicate, so that we can say $\phi_n = \{v \,| C_n(v)\}$.

(iii) $\{\mathcal{R}_k\}$ is a set of rules that apply for the decision. Rules are used to express logical relationships among variables. They are specific evaluations of variables under a given fact. They are suitable for representing inferences or for associating actions with conditions under which actions should be performed [PW12].

Each single rule describes a relationship between a condition and a consequence linked by the statements IF-THEN-ELSE. Let us recall the specification of the rule in the Ontology of the previous chapter, $r_k = \langle A_k, S_k, L_k, W_k, B_k \rangle$ is a rule specification, where A_k denotes the antecedent clauses, Q_k and L_k the consequent actions of the rule, W_k the weight of the rule such that $W_k \in [0, 1]$, and B_k is an auxiliar parameter. Let us extend the notation of the previous chapter, $M(A_k, \mathbb{S}_t)$ denotes the matching predicate, which is true when the antecedent A_k of a rule matches the values of a SOEKS $\mathbb{S}_t$.

(iv) $\{\mathcal{F}_p\}$ is a set of functions that evaluate variables. Functions describe associations between a dependent variable and a set of input variables. Abusing the notation, we can say that $\mathcal{F}_p : \times_{n \in N} \mathbb{V}_n \to \mathbb{V}_p$, that is, $v_p = \mathcal{F}_p(v_{n_1}, \ldots, v_{n_N})$, where $\mathbb{V}_n$ denotes the range of values of variable $\mathcal{V}_n$. Functions can be applied to reduce ambiguity between the different possible states of the variable set and to reason optimal states.

A sequence of SOEKS on the same decision category d, indexed by their time of ocurrence, are stored, together with the corresponding final decision f_t carried out by the decision maker, in a Decisional Chromosome (DChromosome) $\mathbb{C}_d = \{(\mathbb{S}^{\mathrm{t}}, f_t)\}$. A Decisional DDNA (DDNA) is a collection of DChromosomes $\mathbb{D}_m = \{\mathbb{C}_d\}$, which is specific for each decision maker m.

5.3 Experience acquisition process

In the context of this Thesis, a Decisional Event represents a decision made for an individual I_i and a decision domain d_i, for which a set of recommendations have been generated based on a given set of rules. The action of making the decision implies the selection of a final decision $f \in \mathbb{F}$ by the decision maker m_i. Such final decision can be made according to the provided recommendations or not. In a very general setting, a Decisional Event is stored into a SOEKS as follows:

1. Data associated with I_i is mapped into variables $\{\mathcal{V}_n\}$. Following the specification presented in Chapter 4, such data is stored on datatype properties

of the Ontology O of the knowledge model. Variable values v_n are extracted from the instance layer of the ontology, and constraints $\{\mathcal{C}_n\}$ on the variable ranges, from the conceptual layer of the ontology.

2. Rules applying for decision domain d_i are stored in $\{r_k\}$. When following the Extended Reflexive Ontologies (ROX) approach, those rules will be directly extracted from the ontology.

3. Applying Functions are stored in $\{\mathcal{F}_p\}$.

The set of SOEKS stored into DChromosomes and DDNA will follow a temporal succession.

Knowledge consolidation

Once Decisional Events are acquired into a SOEKS structure, the information they contain can be used to drive ruleset evolution algorithms. In the next Sections, some of those algorithms will be presented, allowing to (i) gradually and repeatedly correct rules as well as deprecate them reling on the existing experience, and (ii) generate new rules.

In particular, three different algorithms will be presented, two for rule edition/deprecation (i.e. rule weight evolution and fine-tuning of rules), and a case based reasoning algorithm for new rule generation.

Suggested changes on rules resulting from those methods will be provided at a secondary rule set. Such secondary ruleset will then be analysed by a committee of domain experts, that will agree which of those changes to include in the primary ruleset.

Figure 5.1 shows the complete experience acquisition process, including the generation of SOEKS and the evolution of the ruleset.

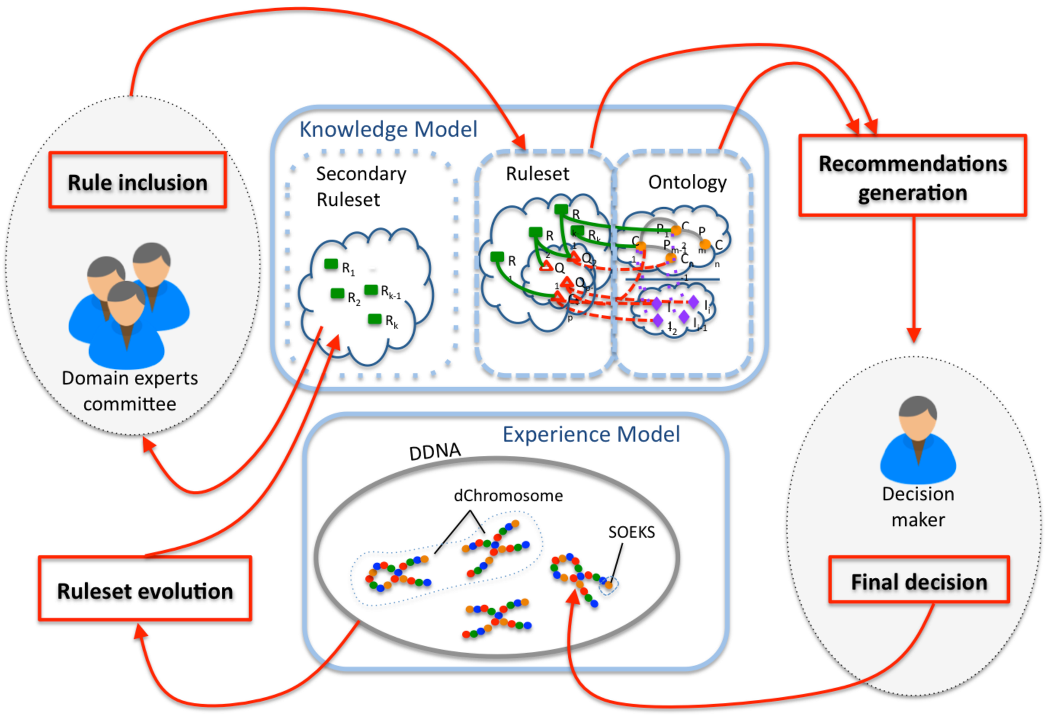

Figure 5.1: Experience acquisition process

5.4 Rule-weight evolution

Rule weight is a core feature of each rule in the SOEKS, used to indicate its importance with regard to other rules that provide recommendations for the same decision. The weight of a rule, W_k, objectively measured, is influenced by three distinct aspects:

1. Quantitative measure: The number of times a rule matches conditions of individuals, and thus its consequent value is recommended by the Decision Support System (DSS).

2. Qualitative measure: The number of times that, when a rule matches conditions of individuals, its consequent value coincides with the final value chosen by the decision maker.

3. Trust/Reputation of decision.

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_t, \ldots, X_T\}$ be a condition matching vector and

$\mathbf{E} = \{E_1, E_2, \ldots, E_t, \ldots, E_T\}$ an error vector, such that

$$X_t = \{x_{t1}, x_{t2}, \ldots, x_{tk}, \ldots, x_{tK_t}\},$$

$$E_t = \{e_{t1}, e_{t2}, \ldots, e_{tk}, \ldots, e_{tK_t}\},$$

where T is the total amount of SOEKS in a dChromosome $\mathbb{C}_d$ and K_T is the total amount of rules in a SOEKS $\mathbb{S}_\mathrm{t}$. Entries x_{tk} and e_{tk} in these matrices have two possible values, 1 or 0, defined as:

$$x_{tk} = \begin{cases} 1 & \text{if } M(A_k, \mathbb{S}_t) \\ 0 & \text{otherwise} \end{cases},$$

$$e_{tk} = \begin{cases} 1 & \text{if } (x_{tk} = 1) \,\&\, (S_k = f_t) \\ 0 & else \end{cases}.$$

Let us define a collection of trust parameters associated to each decision make by decision maker m: $\boldsymbol{\alpha}_m = \{\alpha_{mt}\}$ where each $\alpha_{mt} \in [0, 1]$ is associated to a different decision maker. Let α_{m_t} be the trust of the final decision associated to a $\mathbb{S}_\mathrm{t}$. We define the weight W_k of a rule r_k in the following expression:

$$W_k(\alpha) = \frac{\sum_m \sum_t (x_{mtk} - \alpha_{mt} e_{mtk})}{\sum_m \sum_t \sum_k (x_{mtk} - \alpha_{mt} e_{mtk})}.$$

Algorithm 5.1 contains the pseudo-code of the rule weight evolution algorithm:

5.4.1 Trust

$\boldsymbol{\alpha}_m$ are subjective parameters which measure the perceived trustability of a set of decisions. Trust indicates the level of supervised learning of the process of rule evolution, where a higher α_i implies higher supervision:

(i) When $\alpha_{mt} = 0$: no trust is put on decision maker m, and thus an unsupervised learning is used (quantitative-driven evolution only).

(ii) When $\alpha_{mt} > 0$: the trust level on decision maker m influences the level of supervised learning applied (quantitative- and qualitative-driven evolution).

Each α_{mt} can either be agreed by the team of decision makers, or be set up by every decision maker independently. In the latter case, different weights will be asigned to the same rule, depending on which decision maker sets α_m. The assignment of α_{mt} is done previous to rule weight evolution, and new values can be assigned in the future, if trust on the different decision makers changes.

5.4.2 Recalculation of W_k and convergence of the algorithm

Recalculation of rule weights can be performed either (i) automatically, after a decision occurs or following a certain prestablished frequency such as daily or weekly, or (ii) manually, on demand. When W_k is calculated all SOEKS $\mathbb{S}_t$ for the complete time interval that contain rule r_k are taken into account. Algorithm 5.1 converges towards rule weights that provide recommendations that can be whether:

(i) more frequent, in the case of quantitative-driven evolution only, or

(ii) more similar to the final choices of the decision maker, in the case of quantitative- and qualitative-driven evolution.

5.4.3 Weight zero meaning

Weight zero $W_k = 0$ has two different meanings:

(i) Rule r_k has not matched yet any individuals data.

(ii) Rule r_k has matched some individuals, and at the 100% of the matches the final decision of the decision maker has been different to the recommended.

At the second case, rules r_k such that $W_k = 0$ and $\exists x_{tk} \mid x_{tk} = 1$ are recommended to deprecation.

5.5 Fine-tuning of rules

Fine-tuning of rules consists on adapting rule condition intervals to reduce the difference between recommendations and the final decisions.

Let the antecedent of a rule, A_k , be specified by a set of simple query clauses $q^s_{kl} = \langle V, o, v\rangle$, where V is a variable, o is the comparison operator (i.e. $>, <, =$) and v a value of the range of V. Let us define $M\left(q_{kl}, \mathbb{S}_t\right)$ the matching operator that is true (active) when the simple query clause q^s_l matches the values of a SOEKS $\mathbb{S}_t$. Without loss of generality, we consider only categorical variables V_n. Let us define two parameters (i) μ_{kl}, counting the total amount of times that a query clause is active in a rule that matches conditions, and (ii) ρ_{kl}, counting the total of times that being a query clause active in a rule that matches conditions the final decision of the decision maker is the same as the recommendation of the rule consequent S_k:

$$\mu_{kl} = \sharp\left\{\mathbb{S}_t \mid M\left(A_k, \mathbb{S}_t\right) \& M\left(q^s_{kl}, \mathbb{S}_t\right), q^s_{kl} \in A_k\right\},$$

$$\rho_{kl} = \sharp\left\{\mathbb{S}_t \mid M\left(A_k, \mathbb{S}_t\right) \& M\left(q^s_{kl}, \mathbb{S}_t\right) \& \left(S_k = f_t\right), q^s_l \in A_k\right\}.$$

We define error prone query clauses as those having an error rate $e_{kl} = \frac{\rho_{kl}}{\mu_{kl}}$ greater than a threshold θ. Error prone query clauses are recommended for revision, by a domain experts committe that will decide whether to keep them, change them or remove them. Particularly, query clauses with error rates equal to 100% are recommended for deprecation.

Algorithm 5.2 contains the pseudo code of the rule fine tuning algorithm:

5.5.1 Evolution activation and convergence of the algorithm

The process of fine tuning of rules is activated (i) automatically, after a decision occurs or following a certain prestablished frequency such as daily or weekly, or (ii) manually, on demand.

When fine tuning is calculated all SOEKS $\mathbb{S}_\mathrm{t}$ since last change of rules are taken into account.

Algorithm 5.2 converges towards rules that provide recommendations more similar to the final choices of the decision maker.

5.6 Rule generation

To generate new rules we propose to follow a case based reasoning approach. Let $\{V_s\}$ be the set of variables that are relevant for a decision stored in SOEKS $\mathbb{S}_t$, such that V_s is a variable included in query clauses of A_k and $M(A_k, \mathbb{S}_t)$. Every time a final decision is made decision makers are asked to validate the set of $\{V_s\}$. They are asked to include the variables V_s that they considered during decision making and to remove the non-relevant ones, generating a new set of relevant variables $\{V_s'\}$. Changes in $\{V_s\}$ mean that the recommendations generated the Decision Support System are not complete. Thus, we generate a new rule where the antecedent equals to the values contained in the new set of relevant variables $\{V_s'\}$ and the consequent equals the final decision f (generated rules are of type IF/THEN).

5.6.1 Rules post-processing

The generation of new rules is performed on a secondary ruleset. They are introduced on the ruleset of the Decision Support System (DSS) when analysed by a committee of domain experts, that will agree which of those rules to include. Thus a post processing of the generated secondary ruleset is needed, in order to detect:

(i) spureous rules

(ii) rules already included in others

(iii) rules that generate inconsistencies

Such postprocessing is done before the analysis of the domain experts committee.

5.7 Decision traceability

Traceability of Decisional Events is the ability to keep track of the impact of a decision in the final outcome.

Let us extend the specification of a rule with an objective O_k, such that

$$r_k = \langle A_k, S_k, L_k, W_k, B_k, O_k \rangle$$

as well as the specification of a Decisional Chromosome (DChromosome) with an objective O_t and an outcome U_t,

$$\mathbb{C}_d = \left\{ \left(\mathbb{S}^{\mathrm{t}}, f_t, O_t, U_t \right) \right\},$$

Rules $\{\mathcal{R}_k\}$ applied in the Decisional Event at time t for the generation of recommendations will only be those, such that $\mathcal{R} = \{\mathcal{R}_k \mid (O_k = O_t)\}$.

Measuring the difference between the final outcome of a decision and its objective provides a quality measure of the Decisional Event

$$Q_t = U_t - O_t$$

Q_t can be applied to drive the aforementioned learning methods.

5.8 Discussion - Measurement of the final outcome of a decision

We have presented in this chapter an approach that is oriented towards the formalization and acquisition of Decisional Events from physicians on daily basis. Based on these ideas, we have proposed three different experience-driven learning processes, which evolve a ruleset of a Semantically Steered Clinical Decision Support System (S-CDSS).

Such evolution allows the discovery of new knowledge in the system (intrinsic Knowledge)

i) facilitating the evaluation of the decisions made previously and the analysis of the actions followed, in order to improve the perfomance at a clinical, ethical or economical aspect, and

ii) suggesting new knowledge that could be validated, driving clinical research activities or trials. In this sense, our approach could foster research activities of the medical team.

The discovery of new knowledge combined with decision traceability provides, in addition, a powerful learning tool to learn from erroneos decisions and improve on the future. However, implementation of decision traceability involves several challenges:

i) Temporality should be considered in the system. Nevertheless, our semantic model is relational (context-based) and thus cannot be expressed as time-space. Thus, an auxiliary structure to order the elements in time is required.

ii) By nature, the extraction of the final outcome of a decision may not always be possible.

iii) Additionally, the outcome may not be ready or available at the time when the decision is made. It may be temporally distant (i.e. years far away) to the final decision and operatively the extraction of the outcome may then not be possible.

iv) The outcome of a single decision may not be separable from the outcome of the rest of decisions made over the same individual.

Algorithm 5.1 Pseudocode specification for rule weight evolution algorithm.

```
denominator=0
for t=1 to Number of soeks
{
    Set α_t
    for k=1 to Number of rules
    {
        if conditions of r_k in SOE_t match then
        {
            x_tk=1
            if r_kn.finalDecisionValue equals r_kn.consequenceValue then
            {
                e_tk=0
            }
            else
            {
                e_tk=1
            }
            denominator=denominator+(x_tk + α_t·e_tk)
        }
        else
        {
            x_tk=0
            e_tk=0
        }
    }
}
for k=1 to Number of rules
{
    numerator_k=0
    for t=1 to Number of soeks
    {
        numerator_k =numerator_k + (x_tk- α_t·e_tk)
    }
    weight_k =numerator_k/denominator
}
```

Algorithm 5.2 Pseudocode for rule clause evolution

```
Set $\theta$
for $\mathbb{S}_t$=1 to Number of SOEKS
{
    for $k$=1 to Number of rules
    {
        if $M(A_k, \mathbb{S}_t)$ then
        {
            for $l$=1 to Number of query clauses in rule $k$
            {
                if $M(q_{kl}, \mathbb{S}_t)$ then
                {
                    $\mu_{kl} = \mu_{kl} + 1$
                    if $S_k \neq f$ then
                    {
                        $\rho_{kl} = \rho_{kl} + 1$
                    }
                }
            }
        }
    }
}
for $\mathbb{S}_t$=1 to Number of SOEKS
{
    for $k$=1 to Number of rules
    {
        for $l$=1 to Number of query clauses in rule $k$
        {
            $e_{kl} = \frac{\rho_{kl}}{\mu_{kl}}$
            if $e_{kl} > \theta$ then
            {
                if $e_{kl}$=1 then
                {
                    Recommend deprecation of $q_{kl}$
                }
                else
                {
                    Recommend revision of $q_{kl}$
                }
            }
            else
            {
                Recommend no revision of $q_{kl}$
            }
        }
    }
}
```

Chapter 6

A general architecture for Semantically Steered CDSS (S-CDSS)

Clinical decision support systems (CDSS) are useful tools for physicians, in the context of generating decisional recommendations during clinical tasks. Although different architectures exist for CDSS and the challenges to be bridged are broadly identified, the lack of a semantic perspective in most of them is not to be dismissed. Within the research conducted in this Thesis, we have hypothesised and proved that the addition of semantics would bridge to a certain extend most of the reported difficulties. Nevertheless a plan of action for implementing the so-called Semantic Steered Clinical Decision Support Systems (S-CDSS) is yet to be described.

In this Chapter we present a generic architecture for S-CDSS and make an effort to describe how the challenges are overcome. The aforementioned architecture is applicable in the context where decisional recommendations are to be provided to a physician (e.g. suggestions about the different options available to make a decision). The presented architecture is a natural evolution of architectures previously introduced in [STC$^+$11b, TSC$^+$12]. The evolution is the result of increasing the semantic complexity of the system from a mere knowledge management environment towards an experience management context.

This Chapter is structured as follows: Section 6.2 introduces clinical tasks, proposing a new Clinical Task Model (CTM) where CDSS play an important role. Section 6.3 proposes a new architecture for Semantically Steered CDSS (S-CDSS). Finally, Section 6.5 discusses the way in which the system has bridged CDSS challenges.

6.1 Historic evolution of our architecture

During the the research conducted in this thesis, we have developed 3 different generations of CDSS architectures that responded to the realities of the research projects in which were developed. We call those architectures generations as they evolved into the generic architecture presented in this chapter that would embed and complement them into a wider-more expanded conceptualization.

1st generation: Knowledge reutilization and handling 1st generation architecture presented in [STC+11b] was oriented to the handling and of large amounts of knowledge and re-use of CDSS already in existence inside a given organization. We proposed to semantically enhance clinical data, providing a knowledge model on which reasoning processes could be driven. In our approach, reasoning was supported non solely over explicit knowledge, but also over implicit knowledge.

A four layers scheme was proposed as follows: (i) Data Layer, (ii) Translation Layer, (iii) Ontology and Reasoning Layer, and (iv) Application Layer. The Translation Layer played a key role, by retrieving the corresponding information from the DB of the Data Layer and mapping its data structure into the Knowledge Bases in the Ontology and Reasoning Layer. The main feature of the Translation Layer is the allowance of a decentralized data repository alignment which provides input to a centralized knowledge repository. As a result of the aforementioned feature, DB do not need to intercommunicate directly, modularizing the result. The Ontology and Reasoning Layer (ORL) , deals with the knowledge embedded in the system, to perform reasoning processes that result in utterly recommendations. We proposed three separate modules for the ORL : (i) an ontologies module, where we contributed with a tripple approach formed by the mappings of SWAN, SNOMED CT and a master ontology develped within the project scope (domain), (ii) a query system, where the use of Reflexive Ontologies was implemented, and (iii) a reasoning system, based on production rules. Finally, the Application Layer held the interaction between the user and the system, by means of a graphical user interface (GUI). Figure 6.1 depicts an overview of the 1st generation architecture.

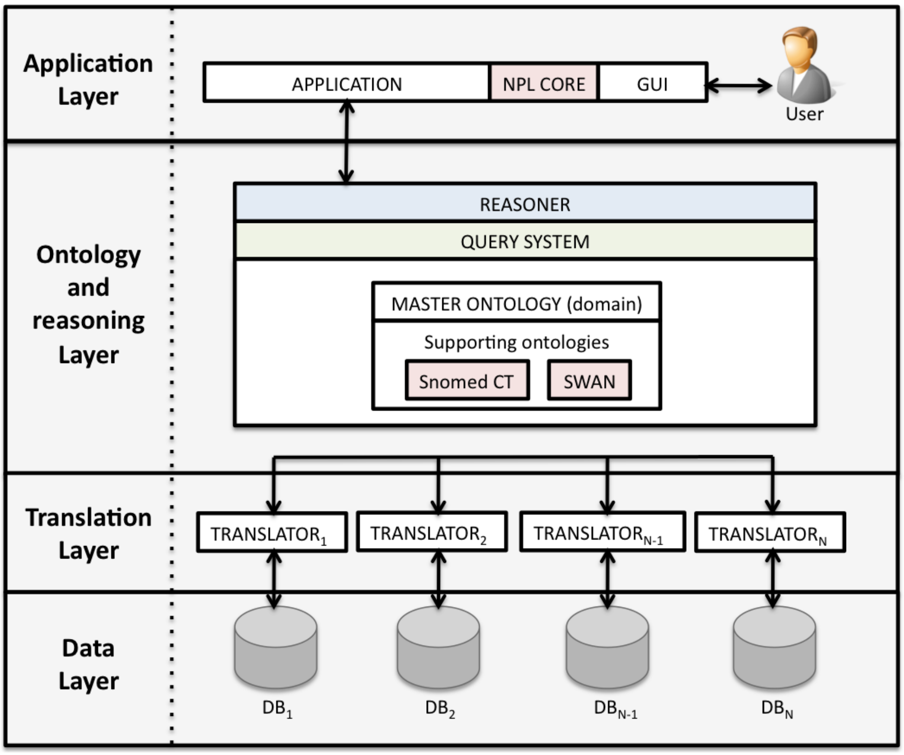

Figure 6.1: Proposed architecture for the Clinical Decision Support System

2nd generation: Experience handling and new knowledge discovery Based on the first architecture, we proposed in [TSC+12] an architecture that was intended for learning from daily-tasks-acquired experience. For the aforesaid purpose an additional **Experience Layer** between the Application Layer and the Ontology and Reasoning Layer was introduced. In such layer, the acquisition, storage and handling of decisional events was carried out.

This architecture was centered on the discovery and maintainability of new knowledge in the system. The domain of application used for testing purposes was the early detection of Alzheimer's Disease (AD). We have shown also that our architecture is valid for supporting (i) diagnosis and daily clinical practice of patients suffering from AD, and (ii) scientific research activities on the causes for AD diagnosis.

Nevertheless, in order to answer other current challenges of decision support mentioned in Chapter 3 a different structure was required, supporting modularity, scalability and the integration of such architecture in the clinical workflow. In the following Sections we present the 3rd generation of our architecture.

6.2 Clinical tasks and decision making

Clinicians perform a series of tasks during the health care process. Such process includes diagnosis, prognosis, treatment, follow-up and prevention. Amongst these tasks, desicions are to be made, and further integrated within the clinical practice workflow. In order to understand the process of decision making, details from the different tasks need to be analyzed and the role of decision makers must be distinguised.

- **Diagnosis** is the process that identifies the syndrome or the disease of the patient [BS03]. Often, each symptom can refer to multiple causes, therefore physicians need to narrow the possibilities. In order to do so, physicians first gather the clinical history, which consists of asking the patient about (i) relevant details of the symptoms of the disease, (ii) past medical history, (iii) family history, and also (iv) habits related to work and leisure. Physical explorations (i.e. auscultation, palpation, inspection and olfaction) are then performed by physicians to detect the signs of the disease that patients cannot report. Finally, complementary explorations such as functional tests, image-based diagnostic tests, endoscopies, biopsies, laboratory tests, and electrocardiography tests, increase the precision of the diagnosis.

- **Prognosis** is the process of generating forecasts about the future evolution of the pathology that affects the patient, such as life expectancy, total or partial functional recovery related to treatment, and future complications [BW11, DC06, Roz06].

- **Treatment** is the process where the overall interactions of a diagnosed disease with patient peculiarities, which can be physical, psychical, economical and social, need to the understood for the prescription of the appropriate therapeutic resources, i.e. hygienic, dietetic, pharmacologic, physical, surgical, psychological [Roz06, DC06].

- **Follow-up** is the process where the effects of treatment and recovery processes [DC06, Roz06] are controlled for guaranteeing a correct evolution of patients. Chronic disease management is particularly focused on this stage.

- **Prevention** is the process that focuses on the avoidance of a certain disease [Gér08]. Prevention can be oriented towards immunization, vaccination and health education, amongst others [Roz06].

In general terms, the different clinical tasks follow a temporal sequence (i.e. prescription of a treatment takes place after a diagnosis is made, then the evolution of the patient can be tracked, etc.). Hence, the whole set of tasks is cyclic by nature instead of lineal, as suggested by Ortiz-Fournier et Al. [OF10]. This fact is due to the strong dependence of physician decisions on (i) the knowledge available at the time of decision, and (ii) the sensitivity of the conclusion to newly gathered evidence by new clinical test results [Roz06]. Figure 6.2 depicts an example of this kind of cyclic processes where flu is not correctly diagnosed at the begining and through a second cycle will be correctly indetified.

The cyclic model needs some additional elaboration, because it does not take into consideration how decisions are reached in clinical practice. A clinical decision maker would handle the knowledge involved in the whole clinical workflow, while learning from each and every situation in order to apply the acquired experience in future decisions [Roz06]. This way of handling knowledge and experience creates an urgent necessity for a model able to support reutilization of knowledge among clinical workflow stages. Federated approaches are proposed mimicking the concept of federation in politics, where a sovereign state is characterized by a union of partially self-governing states under a central government [Xin10]. We follow this approach for the management of the generated clinical knowledge. Thus, in this Chapter we introduce a cyclic and federated Clinical Task Model (CTM) for supporting decision making across clinical stages (Figure 6.3). In the CTM, diagnosis, prognosis, treatment, monitoring and prevention are partially independent. Moreover, several decision makers can be present and provide imput to the system. As every decision made is to be stored and handled in a centralized service, the loss of knowledge throught the clinical decision cycle is lessened, plus an aditional benefit is reached in the sense that every stage could reach a decision not only based on the local environment but having a global view, avoiding important time expenditure and resources waste.

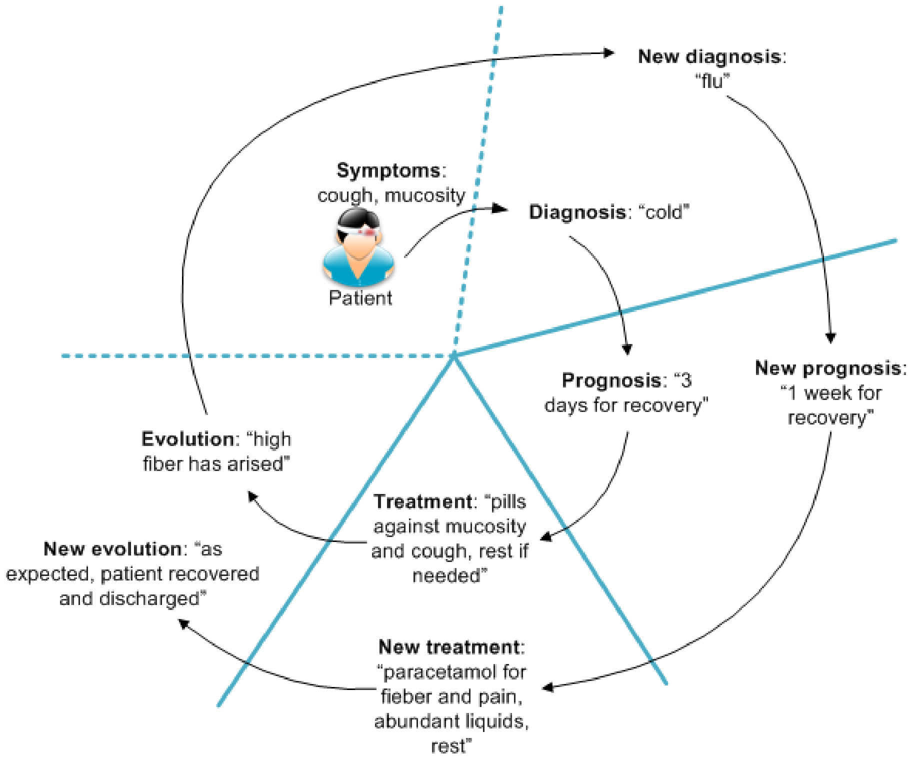

Figure 6.2: Example showing clinical decision making process for a patient apparently suffering from cold first diagnosed as flu.

6.3 The semantic enhancement of CDSS

A CDSS centralized control component handling the knowledge and the performance of the system should support: specialization and control, to cover the different tasks performed during the clinical workflow stages, and knowledge reutilization. Control is implemented in multi-agent systems (MAS) [Ise10] by inter-agent communication and synchronization. Specialization is adequately supported by MAS, where each agent can be oriented at specific tasks. Lastly, knowledge reutilization is supported by the application of semantic technologies. The concept of Semantically Steered CDSS (S-CDSS) originates from the concurrent application of those two technologies to CDSS. In the following sections, the different technologies fit into the distinct elements of the architecture for S-CDSS: (i) a data repository, (ii) a knowledge repository, (iii) the collective experience repository, and (iv) a multi-agent cloud architecture, (depicted in Figure 6.4).

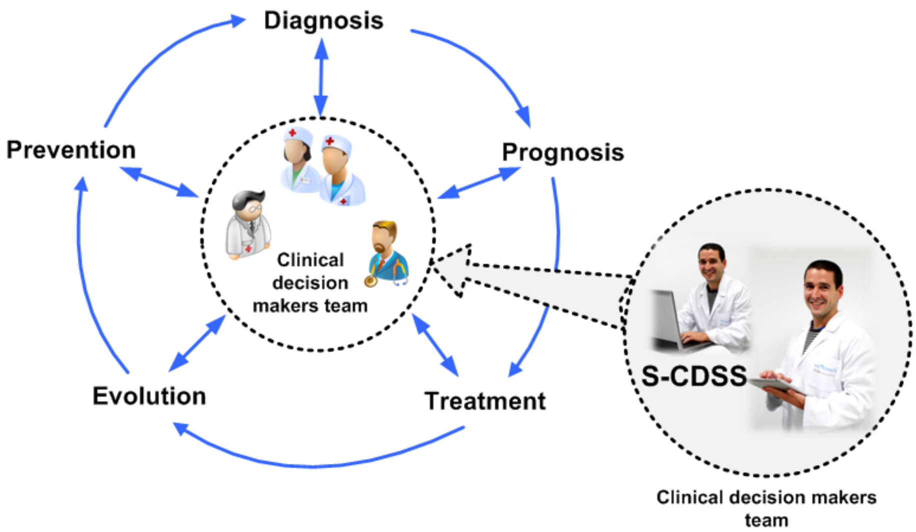

Figure 6.3: Proposed Clinical Task Model and its connection to S-CDSS

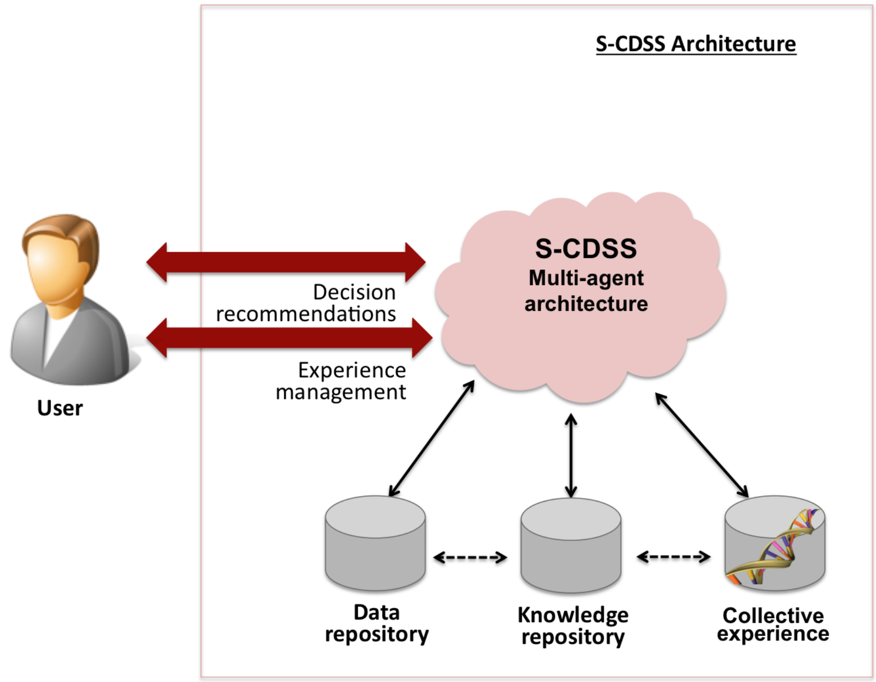

Figure 6.4: Proposed architecture for semantically enhanced clinical decision support

Data repository

The data repository is the source of information for the collection of all decision processes, the content of variables $\{V_n\}$ used by the reasoning and experience gathering processes are extracted from patient data located in the data repository, consisting of:

(i) A set of databases (DB), including clinical systems, electronic health record (EHR) repositories, medical image repositories, picture archiving and communications systems (PACS), and drugs & interactions DB, and

(ii) A set of data sources, like physiological signal acquisition devices (ECG, EEG, respiration rate and effort, spirometry, oximetry, temperature) or other patient monitoring devices. In particular, the different data bases and sources in the repository may be heterogeneous in terms of serialization formats, communication protocols, size, implemented security levels, and location.

The data processing needed to obtain the decision variable values can be rather sophisticated, and it is outside the scope of the Thesis. We assume the existence of such processes as a kind of virtual filter on the data repository. Accessibility to data repository by the S-CDSS needs to be guaranteed, both at a technical and a legal level.

Knowledge repository

The knowledge repository consists of

(i) The ontology O describing all required classes C and properties P of the domain of the S-CDSS, and also containing the instances I extracted from variables $\{V_n\}$ of the data repository.

(ii) The ruleset R containing rules that express relationships between a set of instances and associated actions.

Ontology

The domain ontology of the S-CDSS is built from two sources: on the one hand, extracting the underlying knowledge model from the clinical user experts, and on the other hand, from the alignment with standard ontologies used in the medical domain, such as ontologies in Bioportal [Whe11].

In particular, we hypothesize that the underlying ontologies of a S-CDSS should be the mapping of at least three types of ontologies:

(i) an upper ontology for the medical domain that describes the general clinical domain in a standard way (diseases, procedures, patient parameters, etc),

(ii) an ontology for representing bibliography and the sources for the clinical rules, and

(iii) domain ontologies describing in detail the domain of the current S-CDSS under development.

In our work we have implemented such approach developing for each application domain our own domain ontologies (e.g. the **MIND Ontology**, presented in Chapter 7 for early diagnosis of Alzheimer's Disease, and the **Life Ontology**, presented in

Chapter 9 for diagnosis, treatment and follow-up of Breast Cancer), and aligning the rest two type of ontologies with the following two [STC+11b, STC+11a, TSC+12, STA+12]:

SNOMED CT, as the upper standard ontology. SNOMED CT stands for Systematized Nomenclature of Medicine Clinical Terms, and it is a comprehensive clinical terminology that provides clinical content and expressivity for clinical documentation and reporting. It provides the core general clinical terminology for the Electronic Health Record (EHR) and describes different clinical concepts such as diseases and procedures in a standard way. SNOMED CT is used for standardization purposes, for instance when integrating a newly developed ontology (domain ontology) with a semantized standard, the reutilization of the domain ontology by third party organizations is possible [TGP+09]. In our apporach, we use SNOMED CT to map variables $\{V_n\}$ of the system with standardized terms.

SWAN, as the bibliographic ontology. SWAN is the acronym of Semantic Web Application in Neuromedicine. This effort is the result of a project intended for developing an integrated scientific knowledge infrastructure applied to Alzheimer's Disease (AD) steered by Semantic Web technologies. This ontology is published in the Alzheimer Research Forum website[1], and constitutes a framework for integrating the scientific advances made within different projects and locations in the aforesaid domain. Although AD research was the primary focus in its development, the usability in other biomedical domains is possible [CWW+08]. SWAN links and endorses the criteria of a system with the hypotheses and publications that are being produced by the medical and scientific community [LMC+06]. We take advantage of SWAN for linking rules with their corresponding bibliographic sources.

Rules

Let us recall the specification of a rule in Chapter 4: $r_k = \langle A_k, S_k, L_k, W_k, B_k \rangle$, where A_k denotes the antecedent clauses, Q_k and L_k the consequent actions of the rule,

[1] Alzforum Home Page: `goo.gl/75Dz9N`

W_k the weight of the rule such that $W_k \in [0, 1]$, and B_k is an auxiliar parameter. In the medical domain, rules must be accompained by scientific evidence, and thus, let B_k be the set of bibliographic references that endorse the contents of r_k by the use of SWAN.

A first ruleset is generated by a domain experts committe. External medical knowledge sources such as The Cochrane Library[2] and Pubmed[3], could be used as knowledge sources for domain experts. The use of knowledge extractors and parsers, could directly provide production rules. The resulting ruleset will be maintained and evolved in our approach, by the experience-based process described in Chapter 5.

Collective experience repository

Experiences gathered from the different medical professionals and users of the system are stored in the collective experience repository in the form of SOEKS. The generation of the different Decisional DNAs of the S-CDSS for each different decision maker, and the acquisition process of each SOEKS are described in Chapter 5.

6.4 Multi-agent architecture

Multi-Agent Systems (MAS) combine many autonomous software agents to solve large problems that are beyond the individual capabilities or knowledge of each agent [Ise10, FM99]. MAS are defined by four main characteristics: (i) each agent has incomplete capabilities to solve a problem; (ii) there is no global system control; (iii) data is decentralized; and, (iv) computation is asynchronous [Syc98]. The use of an agent-based paradigm provides the system with the required modularity [Syc98], so that scalability is an intrinsic property of the system. In order to achieve it, the S-CDSS multi-agent architecture supports the inclusion of new agents, that could fulfill, for instance, specific functions belonging to other medical sub-domains, so that their inclusion would broaden the decision support services offered by the architecture and its domains of application. In particular, it consists of eight distinct

[2] The Cochrane Library Home Page: `goo.gl/TDHXAx`

[3] Pubmed Home Page: `goo.gl/VhR9Hq`

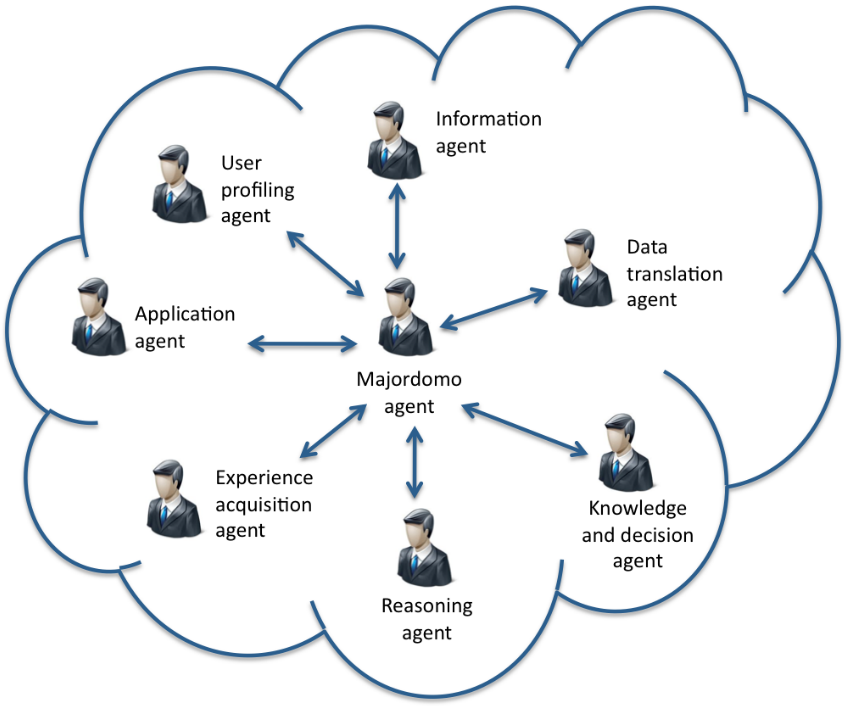

Figure 6.5: Multi-agent cloud architecture

agents, as shown in Figure 6.5: (i) Information agent, (ii) Data Translation agent, (iii) Knowledge and Decision agent, (iv) Reasoning agent, (v) Experience Acquisition agent, (vi) Application agent, (vii) User Profiling agent and (viii) Majordomo agent.

Each agent is defined by five elements:

(i) An univocal identifier,

(ii) the description of the corresponding roles or tasks carried out,

(iii) a status, which can be idle, running, or stopped, and

(iv) a control channel, used by the Majordomo agent to communicate the other agents when to start operations and, if needed, which agent to connect to and how, and

(v) an in/out inter-agent communication channel.

6.4.1 Majordomo agent

The Majordomo agent is in charge of the synchronization and control of the agents in the platform [Cor03]. A blackboard approach [Cra95] is followed, where agents are not allowed to talk directly to each other. All interactions are moderated by the Majordomo. Thereby, security issues are reduced and inconsistencies due to simultaneous communications between different agents are avoided (asynchronism). Whereas the rest of the agents are specialized in different task, the Majordomo agent specialization is the control and performance of the rest of the system.

For each task and agent the Majordomo agent keeps the control by communicating at least twice, during initialization and finalization notification.

6.4.2 Application agent

The Application agent is in charge of the interaction between the user and the system, that will be carried out through a graphical user interface (GUI) dedicated to different purposes: (i) login and user profiling, (ii) data introduction or edition, (iii) authoring tools for the edition or visualization of the underlying models, (iv) decision support, and (v) experience handling.

Visual analytic techniques can be applied to facilitate the visualization of patient data, criteria for decision, next steps on the process, most probable diagnosis, or suitable treatments for a specific patient, among others. The main objective of the Application agent is to facilitate the work of clinicians, and to increase the acceptability of CDSS for their inclusion in the clinical workflow.

The Application Agent initializes every task performed in the system, as actions are requested directly by users. Thus, we also present in this section the diagrams of the complete processes by each of those tasks involving other agents and elements of the architecture.

(A) Login and user profiling

(A1) Task: User login

1. **Get User (Application agent)**

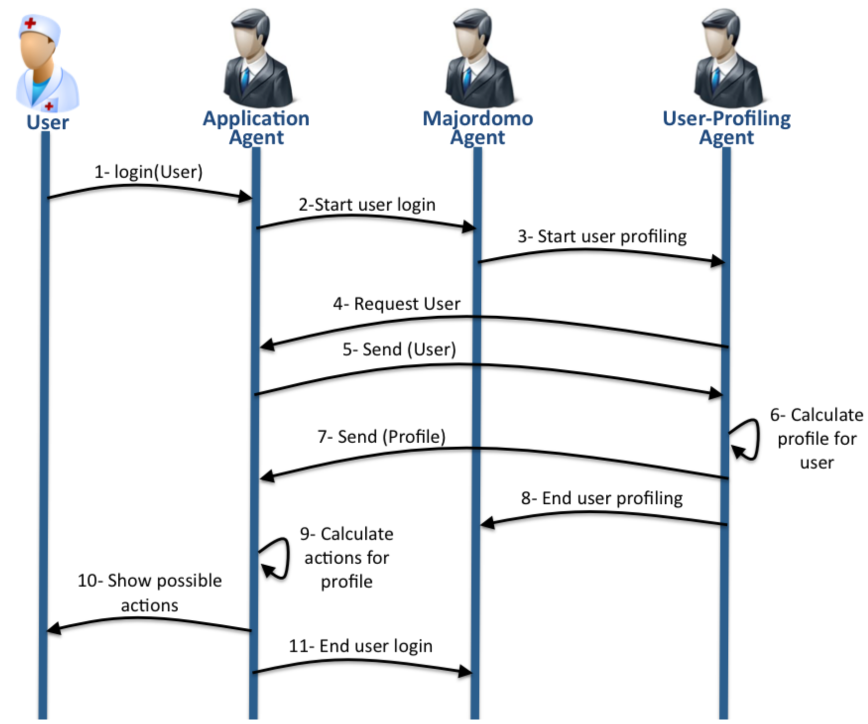

Figure 6.6: User login protocol in the Multi-Agent S-CDSS.

2. Initialization to Majordomo agent (Control Channel)
3. Get (User) request From User-profiling agent (Communications Channel)
4. Send User to User-profiling agent (Communications Channel)
5. Get Profile From User-profiling agent (Communications Channel)
6. **Calculate Actions for user Profile (Application agent)**
7. **Show possible Actions to user (Application agent)**
8. Notify end of task to Majordomo agent (Control Channel)

(B) Data introduction or edition

(B1) Task: Data Editor

1. **Get Data and Action (add, edit, delete) (Application agent)**
2. Initialization to Majordomo agent (Control Channel)
3. Get (Data, Action) request from Information agent (Communications Channel)

4. Send (Data, Action) to Information agent (Communications Channel)
5. Notify end of task to Majordomo agent (Control Channel)

(C) Authoring tools for the edition or visualization of the underlying models

(C1) Task: Knowledge Element Editor

1. **Get Knowledge Element (class in ontology, property in ontology, rule) and Action (add, edit, delete) (Application agent)**
2. Initialization to Majordomo agent (Control Channel)
3. Get (Element, Action) request from Knowledge and Decision agent (Communications Channel)
4. Send (Element, Action) to Knowledge and Decision agent (Communications Channel)
5. Notify end of task to Majordomo agent (Control Channel)

(C2) Task: Knowledge Visualization

1. **Get Element type (ontology, rule) visualization request (Application agent)**
2. Initialization to Majordomo agent ("start knowledge retrieval") (Control Channel)
3. ACK that Knowledge and Decision agent is ready from Majordomo agent (Control Channel)
4. Request (Element) to Knowledge and Decision agent (Communications Channel)
5. Get (Element) from Knowledge and Decision agent (Communications Channel)
6. **Visualize Element (Application agent)**

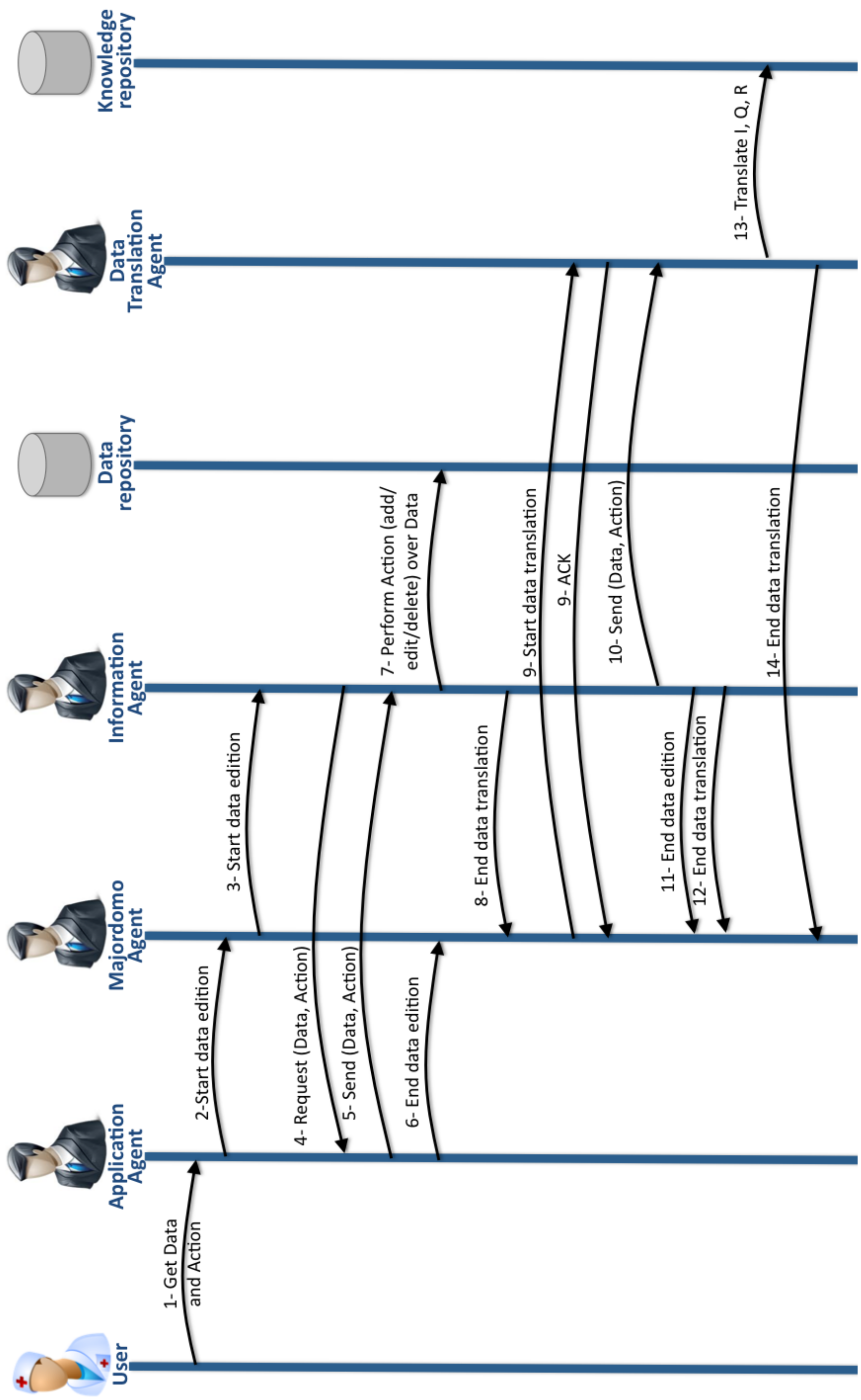

Figure 6.7: Data Edition protocol in the Multi-Agent S-CDSS.

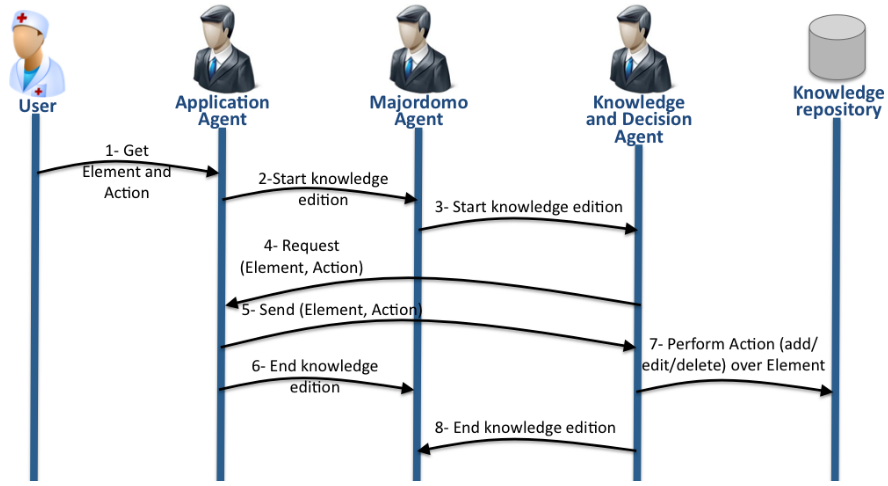

Figure 6.8: Knowledge Edition protocol in the Multi-Agent S-CDSS.

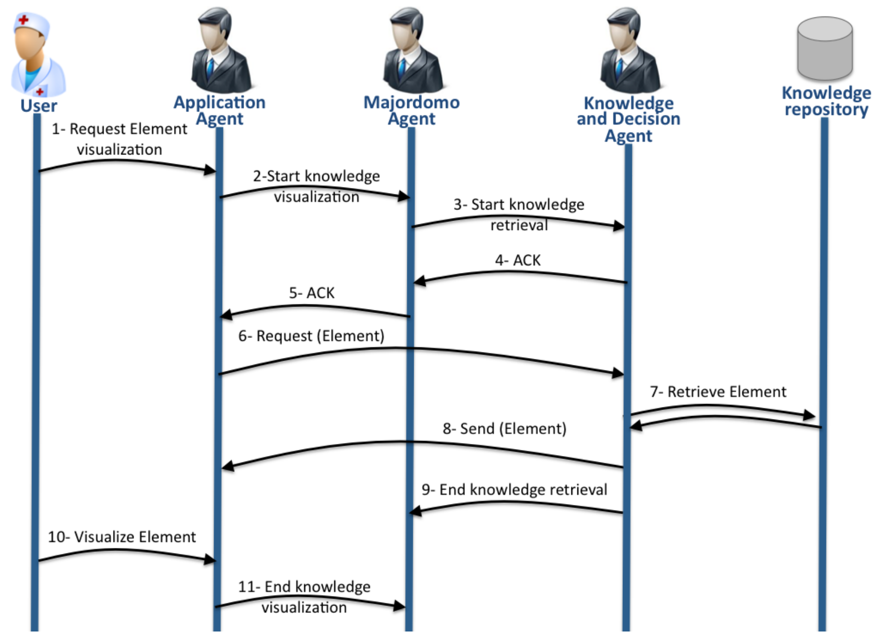

Figure 6.9: Knowledge Visualization protocol in the Multi-Agent S-CDSS.

7. Notify end of task to Majordomo agent (Control Channel)

(D) Decision support

(D1) Task: Get Recommendations

1. **Get request for recommendations of (Individual I_i, Decision cathegory d_i) (Application agent)**
2. Initialization to Majordomo agent (Control Channel)
3. ACK that Reasoning agent is ready from Majordomo agent (Control Channel)
4. Request recommendations (I_i, d_i) to Reasoning agent (Communications Channel)
5. Get recommendations (I_i, d_i) from Reasoning agent (Communications Channel)
6. **Provide recommendations to user (Application agent)**
7. Notify end of task to Majordomo agent (Control Channel)

(E) Experience handling

(E1) Task: Set final decision

1. **Get final decision from user f for (Individual I_i, Decision cathegory d_i, recommendations U) (Application agent)**
2. Initialization to Majordomo agent (Control Channel)
3. ACK that Experience Acquisition and Handling agent is ready from Majordomo agent (Control Channel)
4. Send decisional event (f, I_i, d_i, U) to Experience Acquisition and Handling agent (Communications Channel)
5. Notify Majordomo agent that decisional event was sent to Experience Acquisition and Handling agent (Control Channel)
6. ACK that Reasoning agent is ready from Majordomo agent (Control Channel)

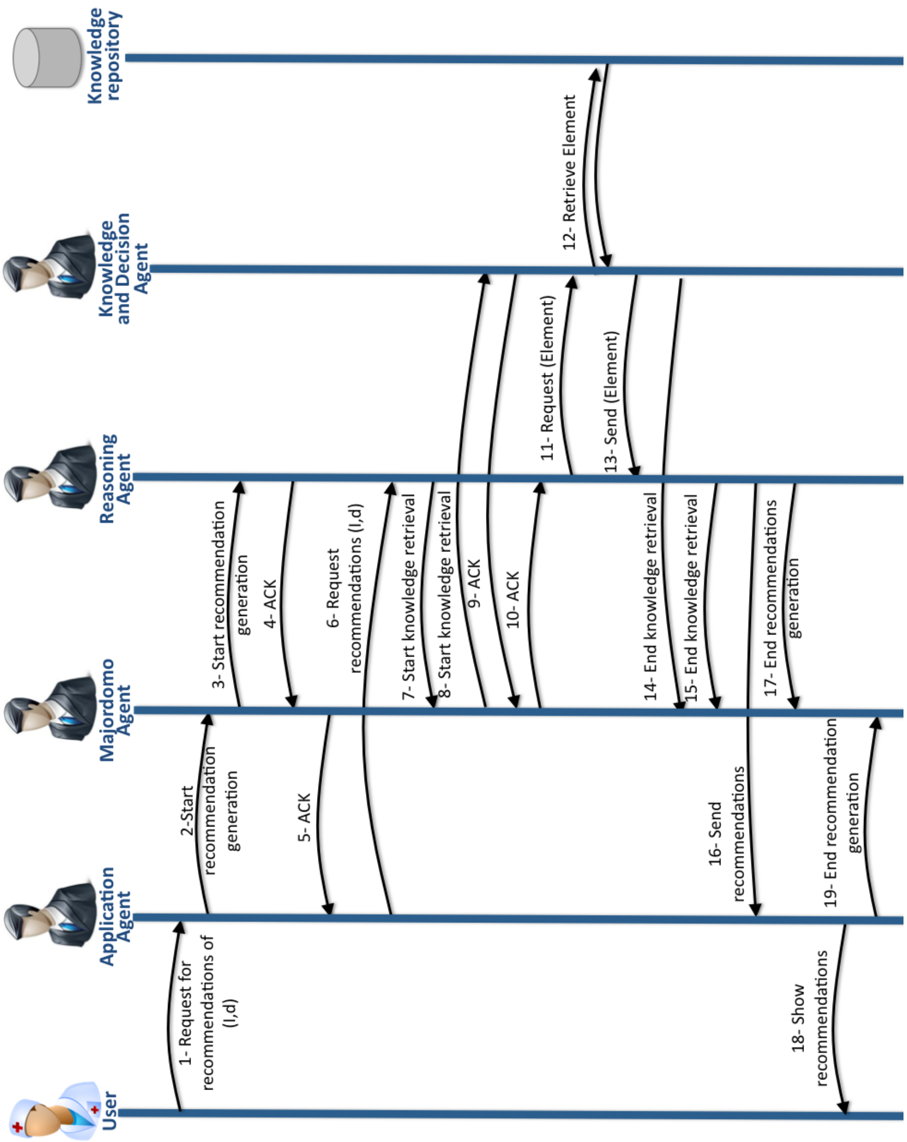

Figure 6.10: Get Recommendations protocol in the Multi-Agent S-CDSS.

7. Request relevant parameters for final decision (f, I_i, d_i, U) to Reasoning agent (Communications Channel)
8. Get relevant parameters for final decision (f, I_i, d_i, U) from Reasoning agent (Communications Channel)
9. **Show relevant parameters to user (Application agent)**
10. **Get updated relevant parameters from user (Application agent)**
11. Send updated relevant parameters for final decision (f, I_i, d_i, U) to Reasoning agent (Communications Channel)
12. Notify end of task to Majordomo agent (Control Channel)

(E2) Task: Ruleset evolution

1. **Get request for ruleset evolution (rule weight evolution, fine-tuning of rules) (Application agent)**
2. Initialization to Majordomo agent (Control Channel)
3. Notify end of task to Majordomo agent (Control Channel)

6.4.3 User-Profiling agent

When a user logs in the system, the User-Profiling agent characterizes the user. Such characterization is performed using the minimum number of parameters that could characterize user behavior and user attributes. Existing user characterization modules such as GOMS [Gra93] and CommonKADS [Has11], present implementation and logic modules that can be used development of such agent.

(A1) Task: User profile calculation

1. Get initialization from Majordomo agent (Control Channel)
2. Request (User) to Application agent (Communications Channel)
3. Get (User) from Application agent (Communications Channel)

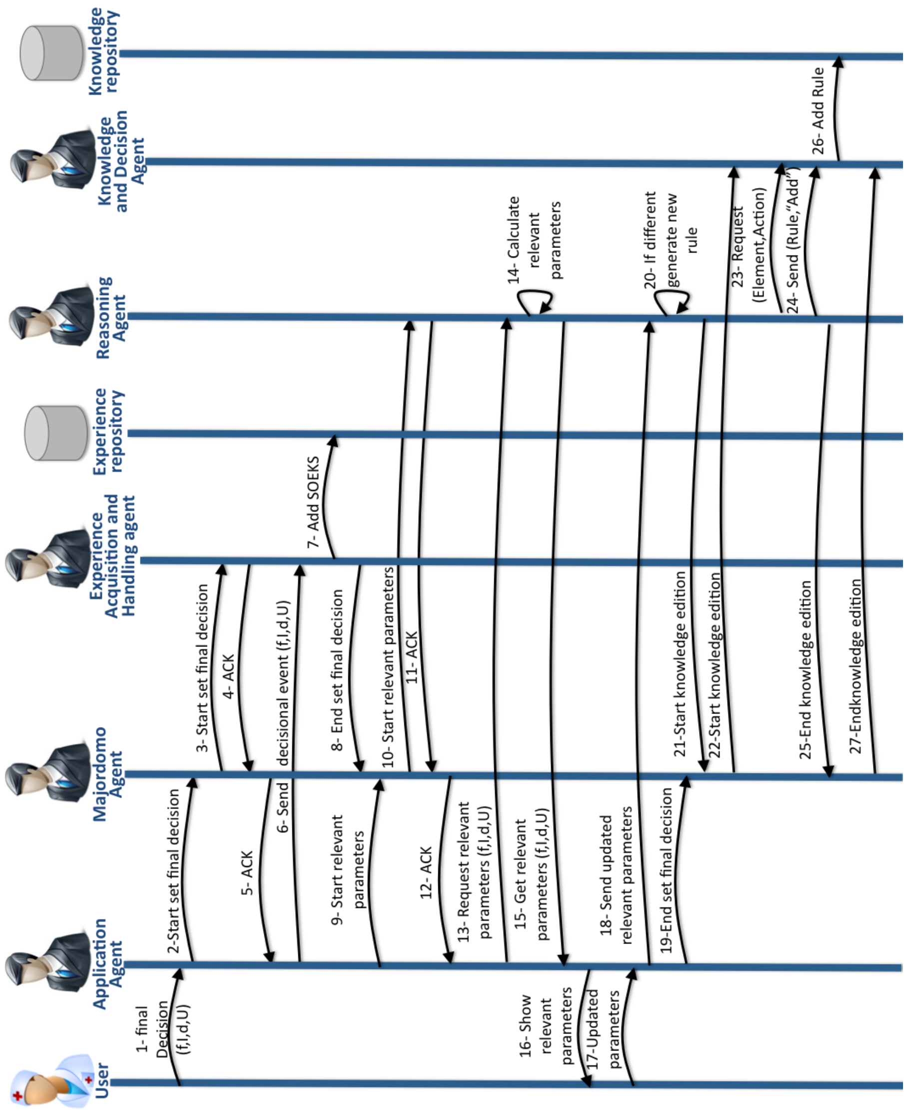

Figure 6.11: Set Final Decision protocol in the Multi-Agent S-CDSS.

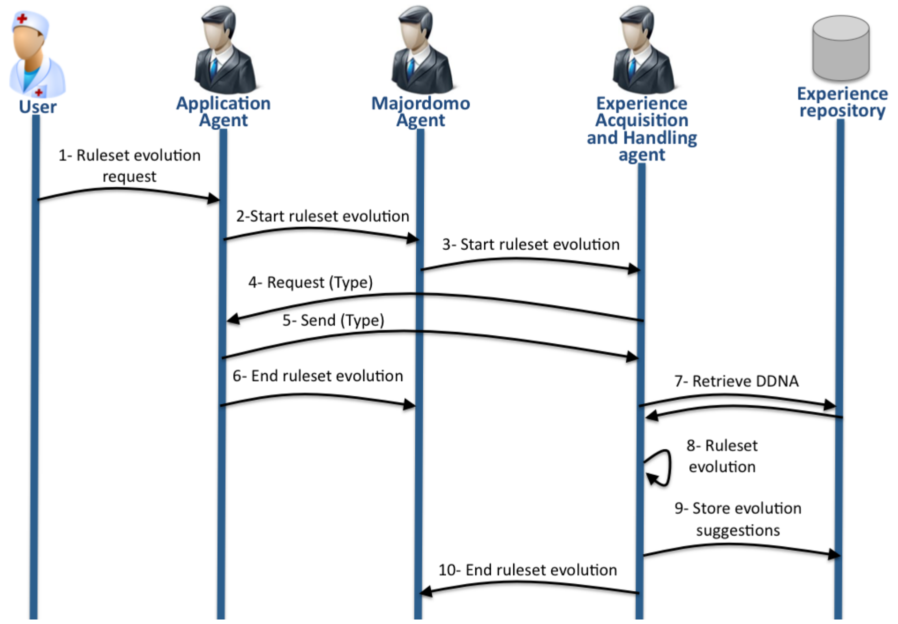

Figure 6.12: Ruleset Evolution protocol in the Multi-Agent S-CDSS.

4. **Calculate Profile for user (User-profiling agent)**
5. Send (Profile) to Application agent (Communications Channel)
6. Notify end of task to Majordomo agent (Control Channel)

6.4.4 Information agent

This agent accesses the information stored in the data repository of the architecture. It is in charge of handling (adding, deleting, and editing) data dealing with the corresponding web services and data accessing language and protocols.

(B1) Task: Data addition, deletion and edition

1. Get initialization from Majordomo agent (set applying agent) (Control Channel)
2. Request (Data, Action) to applying agent (Communications Channel)
3. Get (Data, Action) from applying agent (Communications Channel)
4. **Perform Action (i.e. add, edit or delete) over Data in Data Repository (Information agent)**

5. Initialization of Majordomo agent ("start data translation in Data Translation agent") (Control Channel)

6. ACK that Data Translation agent is ready from Majordomo agent (Control Channel)

7. Send (Data, Action) to Data Translation agent (Communications Channel)

8. Notify end of task to Majordomo agent (Control Channel)

6.4.5 Data Translation agent

The approach for clinical workflow integration presented in the previous section requires that plain information in the system would be semantically enhanced in order to support richer reasoning processes. In order to do so, the Data Translation agent performs the mapping of the data structure in the data repository to the knowledge models in the knowledge repository. Such mapping is performed by relating Variables $\{V_n\}$ of the data repository to Datatype Properties P_i^d of the ontologies in the knowledge model, as well as Instances I_i that take values $\{v_n\}$ in $\left\{P_n^d\right\}$ corresponding to $\{V_n\}$.

(B1) Task: Data Translation

1. Get initialization from Majordomo agent (Control Channel)

2. Notify ACK to Majordomo agent (Control Channel)

3. Get (Data, Action) from Information agent (Communications Channel)

4. **For each data perform Action (i.e. add, edit or delete) over the corresponding Instances I_i and Datatype Properties P_i^d in Knowledge Repository (Data Translation agent)**

5. **(If knowledge base is built by Extended Reflexive Ontologies) Update Queries and Rules that match for Instances I_i in Knowledge Repository (Data Translation agent)**

6. Notify end of task to Majordomo agent (Control Channel)

6.4.6 Knowledge agent

The Knowledge and Decision agent deals with the handling of the knowledge model. Two separate modes are defined: (i) the editor mode, to add, edit or delete (a) classes and properties in the ontology, and (b) rules in the ruleset, and (ii) the retriever mode, to retrieve the ontology or the ruleset from the Knowledge Repository. The Knowledge and Decision agent is responsible of guaranteeing the maintainability and extensibility of the knowledge in the system.

(C1) Task: Knowledge Element Editor

1. Get initialization from Majordomo agent (set linking agent) (Control Channel)
2. Send (Element, Action) request to linking agent (Communications Channel)
3. Get (Element, Action) from linking agent (Communications Channel)
4. **Perform Action (i.e. add, edit or delete) over Element (i.e. Class in Ontology, Property in Ontology, Rule) in Knowledge Repository (Knowledge and Decision agent)**
5. Notify end of task to Majordomo agent (Control Channel)

(C2) Task: Knowledge Element Retrieval

1. Get initialization from Majordomo agent (set linking agent) (Control Channel)
2. Notify ACK to Majordomo agent (Control Channel)
3. Get request of Element (i.e Ontology, Ruleset) from linking agent (Communications Channel)
4. **Retrieve Element from Knowledge Repository (Knowledge and Decision agent)**
5. Send Element to linking agent (Communications Channel)
6. Notify end of task to Majordomo agent (Control Channel)

6.4.7 Reasoning agent

The Reasoning agent interacts with the knowledge model, a semantic reasoning tool and the query engine, in order to obtain inferred responses that will aid clinicians during decision making. In Chapter 4 a formal specification of such process is provided.

(D1) Task: Get Recommendations

1. Get initialization from Majordomo agent (Control Channel)
2. Notify ACK to Majordomo agent (Control Channel)
3. Get request for recommendations of (Individual I_i, Decision cathegory d_i) from Application agent (Communications Channel)
4. Initialization of Majordomo agent ("start element retrieval in Knowledge and Decision agent") (Control Channel)
5. ACK that Knowledge and Decision agent is ready from Majordomo agent (Control Channel)
6. Request Rules related to d_i and Individual I_i (rule retrieval) to Knowledge and Decision agent (Communications Channel)
7. Get Rules related to d_i and Individual I_i from Knowledge and Decision agent (Communications Channel)
8. **Generate recommendations (Reasoning agent)**
9. Send recommendations (I_i, d_i) to Application agent (Communications Channel)
10. Notify end of task to Majordomo agent (Control Channel)

(E1) Task: Generate relevant parameters from a decision and a new rule

1. Get initialization from Majordomo agent (Control Channel)
2. Notify ACK to Majordomo agent (Control Channel)
3. Get request of relevant parameters for final decision (f, I_i, d_i, U) from Application agent (Communications Channel)

4. **Generate relevant parameters from** (f, I_i, d_i, U) **(Reasoning agent)**
5. Send relevant parameters for final decision (f, I_i, d_i, U) to Application agent (Communications Channel)
6. Get updated relevant parameters for final decision (f, I_i, d_i, U) from Application agent (Communications Channel)
7. **If updated relevant parameters are different, then generate new rule (Reasoning agent)**
8. Initialization of Majordomo agent ("start rule addition in Knowledge and Decision agent") (Control Channel)
9. Get (Element, Action) request from Knowledge and Decision agent (Communications Channel)
10. Send ("new rule", "add") to Knowledge and Decision agent (Communications Channel)
11. Notify end of task to Majordomo agent (Control Channel)

6.4.8 Experience Acquisition and Handling agent

The Experience Acquisition and Handling agent gathers and stores the experience of clinicians and other users in the system, providing automatic maintenance and updating of the knowledge model, as explained in Chapter 5. For this purpose, variables, functions, constraints and rules involved in every decisional event are handled.

(E1) Task: Set final decision

1. Get initialization from Majordomo agent (Control Channel)
2. Notify ACK to Majordomo agent (Control Channel)
3. Get decisional event (f, I_i, d_i, U) from Application agent (Communications Channel)
4. **Generate and save SOEKS S_i in Experience Repository (Experience Acquisition and Handling agent)**

5. Notify end of task to Majordomo agent (Control Channel)

(E2) Task: Ruleset evolution

1. Get initialization from Majordomo agent (Control Channel)
2. **Retrieve DDNA from Experience Repository (Experience Acquisition and Handling agent)**
3. **Evolve ruleset (Experience Acquisition and Handling agent)**
4. **Save evolution suggestions in Experience Repository (Experience Acquisition and Handling agent)**
5. Notify end of task to Majordomo agent (Control Channel)

6.5 Discussion - Answering to CDSS challenges

In this Chapter, a Clinical Task Model (CTM) has been presented, locating the different clinical information processing stages (i.e. diagnosis, prognosis, treatment, evolution and prevention). On the basis of the CTM, we have presented a S-CDSS architecture providing a framework for the integration and reutilization of decision support systems in clinical environments, while answering the main CDSS challenges identified. In particular, computerized clinical decision support is provided by a rule system and reasoning engine that infers the corresponding decision recommendations. System extensibility is guaranteed by the use of a Multi-Agent System architecture providing modularity, scalability and reutilization. On the other hand, the acquisition and handling of experience provided by SOEKS/DDNA allows the maintainability of the underlying knowledge bases of the S-CDSS. Additionally, timely advice is provided by the use of Reflexive Ontologies, that speed up reasoning processes and improve the overall efficiency of the system. Lastly, the evaluation of costs and effects of CDSS is also supported by the handling of the acquired experience.

Clinical workflow integration has also been tackled in this chapter, but only at the level of knowledge reutilization amongst the different clinical tasks. Integration of our S-CDSS within clinical systems of hospitals or medical centers is still a challenge. In our approach, we have assumed that variables $\{V_n\}$ can be directly retrieved from data bases and sources. In fact, in both case studies were we have applied S-CDSS (see Chapters 7 and 9), data was provided in a variable-value structure. Hence, loading such data in our system was direct. However, in other clinical environments patient data is stored in Electronic Health Record (EHR), which follows a different structure and a data storage paradigm. Particularly, patient data is generally stored textually, and thus the extraction of $\{V_n\}$ becomes a natural language processing task. In order to extend the use of our system to a more general case, we consider the development of such natural language processing module as our next step.

S-CDSS Architecture Our proposed architecture for S-CDSS can be framed under the cathegory of Service-Model architecture presented by Wright et al. [WS08]. The standard interface in our case is at the side of patient data. If our system is able to extract variables $\{V_n\}$ from those data, the loading and reasoning processes do not need to be done locally. It can instead be provided as a web service, if security and confidentiality issues are solved.

The most important part in such case would be to guarantee that the knowledge model and the extracted data are aligned. For that purpose, a specific knowledge model for each hospital is needed, which is provided by their experts, but at the same time a standard model could gather the rest of the models together. Such gathering can be performed by a social process, where domain experts can agree on the protocols. Therefore, our S-CDSS architecture could lead to a Framework for Social Clinical Guidelines and Protocols.

Part III

Case studies

Chapter 7

Decision support for the early diagnosis of Alzheimer's Disease (AD)

This Chapter presents a case study of our proposed ideas and methodologies in the domain of the early diagnosis of Alzheimer's Disease (AD). This work has been developed under the framework of the Spanish project MIND [1], focused on the multidisciplinary approach of AD. Under the framework of the aforesaid project, a clinical trial over 350 patients and 3 hospitals in Spain was performed. Our group was in charge of the development and implementation of a clinical decision support system explained in deep in previous chapters.

This Chapter is structured as follows: Section 7.1 introduces the technical problem and identifies challenges to be overcome. Section 7.2 summarizes our contributions to the project. Section 7.3 provides a short overview of the state-of-art on current CDSS for the early diagnosis of AD. Section 7.4 proposes a series of Knowledge Bases (KB) including our own original ontology for modeling the domain of the early diagnosis of AD, and a set of production rules applicable to the KB. Section 7.5 presents implementation details of the system. Lastly, Section 7.6 presents a discussion about the needs of intrinsic Knowledge discovery by means of the evolving

[1] Home page of the research project (in Spanish): www.portalmind.es

of production rules, aimed at mimicking a human-like learning paradigm.

7.1 Description of the technical problem and challenges identified

Alzheimer Disease (AD) is a neurodegenerative disorder discovered by the German neurologist Alois Alzheimer in 1906. It is a progressive and cronic disease, whose cause is still unknown. AD is characterized by a slow progression, which starts with memory loss problems and ends up with severe brain damage. The evolution of the disease varies depending on the person, and it is tightly related to aging. During the progress of AD, neurons are damaged and the information flow between them is cut off, so that mental capacities are progressively lessened and some parts of the brain are irreversively degenerated.

Feldman et al. [FW05] describe 3 different phases of AD:

- Mild AD, characterized in most cases by short-term memory loss and spacial or temporal desorientation;
- Moderate AD, where patients lose language fluidity and the hability to execute daily activities alone, and
- Severe AD, where patients are not able to support themselves anymore, they do not recognize relatives or close friends, their muscles become rigid and they can fall in agitation states.

Additionally, Mild Cognitive Impairment (MCI) is reported to be a previous phase to AD, such that all AD patients have initially MCI, but not all MCI patients evolve to AD. Figure 7.1 depicts the progression of AD and its phases, evaluated by the Mini Mental State Exam (MMSE).

It is a well-known fact, that life expectancy of the population in developed countries is increasing. Some reasons of this behaviour are found partly in the medical advances achieved during the last decades and the changes in living standards. As a consequence, AD has become a major issue, affecting 10% of the senior population

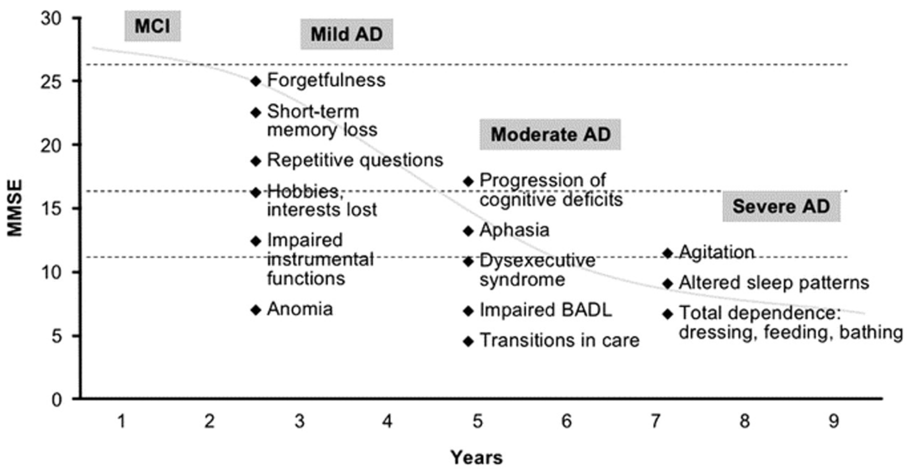

Figure 7.1: Phases of AD, described by Feldman et al. [FW05]

and more than 25 million individuals around the world [Ple10]. Actually, AD is the leading cause of dementia in the aging population [CM10b] and hence the sociological and economical impact of AD in the society is huge [MAB06, SCGR+11]. In Spain the situation is alarming[2]: around 600.000 people have been diagnosed with AD [dPCVOV+09] and it is believed that 200.000 more are already suffering from AD, but have not been diagnosed yet. According to the Spanish Alliance of Relatives of Alzheimer and Other Demency Patients[3], it is estimated that in 2025, 1.200.000 AD patients will live in Spain. The annual cost per patient in Spain is around 18.000 euros, representing a gross cost for the country of around 10.000 million euros. An early diagnosis and the implentation of treatments to delay the evolution of the disease would increase the quality of life of AD patients and might save annualy around 600 million euros. In the aforesaid scenario there is a need of improving the medical attention to AD patients, but the following challenges arise.

Challenge 1: Knowledge discovery

As mentioned before, the cause of AD is still unknown. In this context treatments and prevention measures are quite difficult to develop, and there is currently no cure

[2]Statistics given by AFAGI (Asociación de Familiares de Enfermos de Alzheimer de Gipuzkoa). Home page: http://www.afagi.org/

[3]Home Page of the Confederación Española de Familiares de Enfermos de Alzheimer y otras Demencias (CEAFA): `http://www.ceafa.es/`

for it. Upon diagnosis palliative treatments of the symptoms are the unique therapy available [MAB06,SCGR+11]. Alzheimer patients live an average of 8-10 years after the disease has been diagnosed. For that reason, an early diagnosis when the brain damage is still small is highly desirable [BJB+05].

Nevertheless, recent studies in early detection of AD have identified initial stages as far as 15 years before the first clinically recognizable symptoms appear [MAB06]. Thus, the discovery of new biomarkers and protocols that could lead to such desired early diagnosis is needed, as well as for the identification of individuals at risk. As new technologies and tools are required to support knowledge discovery, this Thesis intends to dive deep into the Knowledge acquisition and reasoning that semantic approaches provide in the hope of bridging such gap.

Challenge 2: Knowledge handling

In the current approach for diagnosis of AD, it is reached thought the analysis of patient-test results carried out during different and multidisciplinary medical tests (e.g. neurological, neuropsychological, and neuroimaging tests) [MAB06]. As the tests generate massive data, large amount of information is generated, and high efforts are required for its analysis. The fact that new medical advances on AD are frequent involve an exponential grow of new knowledge that must be furterly handled and kept up-to-date. Tools easing the handling of such information and knowledge are needed and we believe that again a semantic approach offers a very good path towards the solution of the aforesaid gap.

7.2 Brief description of our contributions

Taking into account the previously reported challenges, we have developed a Clinical Decision Support System (CDSS) for the early diagnosis of AD. The architecture of this system corresponds to the 1st generation of CDSS architectures presented in Chapter 6. Such approach, as mentioned before, was focused on the knowledge layer, which deals mainly with both knowledge discovery and knowledge handling.

In brief our contributions can be summarized as follows:

- We modeled the domain of tests for diagnosing AD. We examinated current domain ontologies and their coverage, concluding that a new domain ontology was needed, containing all medical tests carried out by clinicians during AD diagnosis (in the context of the MIND Clinical Trial). We named that new ontology the MIND Ontology, and mapped it into SNOMED CT and SWAN ontologies.

- We generated production rules for AD diagnosis. We developed the rule set following a dual approach, including decision criteria extracted from the official clinical guideline for demencies of the Spanish Neuroscience Association, as well as criteria coming directly from domain experts (clinical practitioneers).

- We developed a rule based inference engine to infer the corresponding diagnoses. The query system implements the Reflexive Ontologies fast querying technique in order to speed up the reasoning process.

- Our CDSS provides decision recommendations to clinicians about diagnoses. Our recommendations are provided as a support to their decisions, taking special care in not producing any kind of automatic behaviour as the spirit of our system is not to replace clinicians.

- We provide interactive facilities for manual knowledge discovery, where a user can extend and update the knowledge model at any time in the most intuitive possible manner.

7.3 Brief review of current ontologies in the domain of AD

AD is a neurodegenerative disease that requires an eminently multidisciplinary approach for diagnosis. This fact imposes restrictions on CDSS in terms of the set of techniques to be applied. During the last years, quite a few ontologies have been presented in the literature regarding different aspects of the disease, in the following paragraphs, we introduce the most relevant ones:

- The AD Ontology (ADO) presented by Malhotra et Al. [MYG+13] is a disease ontology representing clinical features, treatment, risk factors, and other aspects of the current knowledge in the domain of AD. Its main objective is to develop a semantic framework for an interoperable and standardized representation of the knowledge in the AD domain. ADO by nature, is specially oriented towards an upper represenation of the formalization of terms, rendering its use kind of problematic in a real world scenario. Thus, aligning our work, which is only limited to the diagnosis of AD, with ADO could arguably endorse the modeling of the domain we performed, and will be a future work.

- The Common Alzheimer's Disease Research Ontology (CADRO) presented by Refolo et Al. [RSL+12] is a classification system intended to integrate and analyze different AD research portfolios from public and private organizations working in the domain worldwide. Although the scope and the purpose of this ontology are broader than ours, our work on test modeling could be fitted into Cathegory B of the CADRO Ontology.

- The SWAN project (Semantic Web Applications in Neuromedicine) aims to develop a generic semantically structured framework for biomedical discourse, which has been initially applied to AD research [CWW+08]. Ontologies in SWAN cover different complementary aspects: (i) people, groups and organizations, (ii) discourse elements, (iii) bibliographic records and citations, (iv) life science entities, (v) tags, qualifiers, and vocabularies, and (vi) versions and provenance. In our work, we will directly apply the bibliographic records and citations part of SWAN.

Table 7.1: Description and focus of each of the ontologies applied in the MIND project.

ONTOLOGY	DESCRIPTION	FOCUS
Snomed CT	Patient	Clinical
SWAN	Domain	Symptomatical
MIND Ontology	Tests	Diagnosis

7.4 Modeling the domain knowledge of AD diagnosis

7.4.1 Ontologies for the early diagnosis of AD

We have developed the ontolologies of a CDSS for early diagnosis of AD following the methodology proposed in Chapter 6, which consists on the mapping of three different ontologies: an upper standard ontology for the clinical domain, an ontology for the representation of bibliography and a domain ontology for the early diagnosis of AD. For that purpose, we have mapped the following three ontologies: SNOMED CT, SWAN and our own domain ontology called the MIND Ontology. Figure 7.1 depicts the three ontologies with their usability and context of application.

7.4.2 Brief description of the MIND ontology (domain)

The MIND ontology models the tests carried out on the patients (e.g. neuropsychological, neurological, radiological, metabolomical and genetic). For its development we followed Methontology as our ontology development methodology [FGJ97, GPFC04]. In order to develop this ontology, we had the support of a group of domain experts who were instructed to describe tests usually performed on AD suspected patients[4].

There are seven classes that are automatically mapped during generation time: **Doctor**, **Patient**, **Diagnosis**, **Enrollment**, **FollowUp**, **Test** and **TestValue**. The **Test** class is the superclass of the different tests applied, and in general a

[4]Due to confidentiality issues of the industrial project, in this Thesis we are going to present a brief description of the MIND Ontology instead of discussing in deep its details

new test reaching the system, would be considered as a sub class of this class. The **Test** class is related with the class **Patient** through the *correspondingPatient* property, and with the class **Doctor** through the *orderingDoctor* property and at the same time with the class **FollowUp** with the *correspondingFollowUp* property. The **Diagnosis** and **Enrollment** classes are related to the class **Patient** through *hasDiagnosis* and *hasEnrollment* properties, respectively. The instances of the **TestValue** class are the data gathered in the GUIs. Hence, **Test** and **TestValue** are related to the properties that refer to those parameters in the aforesaid GUIs. In other words, these are results of the different tests carried out and they are mapped to SNOMED CT, by assigning each property the corresponding SNOMED CT code. Figure 7.2 depicts the described ontology.

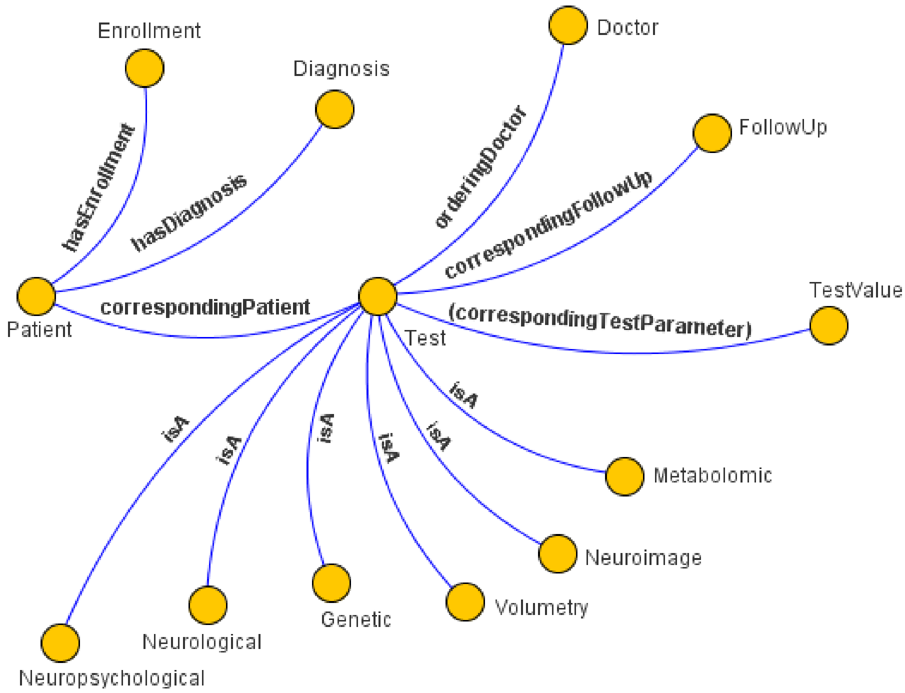

Figure 7.2: Overview of the MIND ontology

7.4.3 Ruleset for early diagnosis of AD

Our ruleset has been generated by domain experts in two different hospitals in Spain, Hospital General de Castilla La Mancha (HGCLM), in Ciudad Real, and Hospital Universitario Virgen del Rocío (HUVR), in Seville. Two different rule generation approaches have been followed: In HUVR the sections about diagnosis of Alzheimer's Disease coming from the Official Dementia Clinical Guideline from the Spanish Neurology Association [AAA+09] were translated to rules. In HGCLM a neurologist provided rules based on their procedures on diagnosis (in the form of reasons and outcomes).

In order to provide a ranking mechanism, we have asked the domain experts to assign a weight W_k to each rule, signifying the importance or precedence for its application during reasoning time. Additionally, due to the reported importance of providing written backup on the causes or events that would generate a rule (evidence checking), every rule is endorsed by a bibliographic resource (by means of a mapping with SWAN [CWW+08]). Figure 7.3 presents an example of one of the production rules. In Appendix A an example rule (for the BReast Cancer domain) is shown, following the same format as in the MIND project for early diagnosis of AD.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RuleSet>
    <LoadRule>
        <Rule>if ( ( CLASS Neurological with the PROPERTY Neur
        <weight>0.6</weight>
        <AccordingTo>doi: 10.1016/S0028-3932(01)00055-0</Accor
    </LoadRule>
</RuleSet>
```

Figure 7.3: Production rule example

7.5 System implementation

7.5.1 Context and environment

The system has been implemented for its use during the MIND Clinical Trial, which aims at detecting early which MCI patients evolve to AD and why. Patients taking part in the study fulfill the inclusion requirements set by our experts and can belong to one of the following 3 groups: Alzheimer, Mild Cognitive Impairment (MCI) or control. Patients are followed up every 6 months during 3 years and each time the entire set of tests is applied to them. To be precise, the tests carried out are divided into the following categories:

- Neuropsychological Tests
- Neurological Tests
- Blood analysis (for the genetic and metabolomical results)
- Magnetic Resonance Imaging (MRI)
- Functional Magnetic Resonance Imaging (fMRI)

7.5.2 System architecture

The architecture developed for the MIND project is a 1st generation architecture as described in Chapter 6. It handles efficiently the tasks of knowledge discovery and knowledge handling. Figure 7.4 depicts the architecture implementation of the CDSS, called ODEI system.

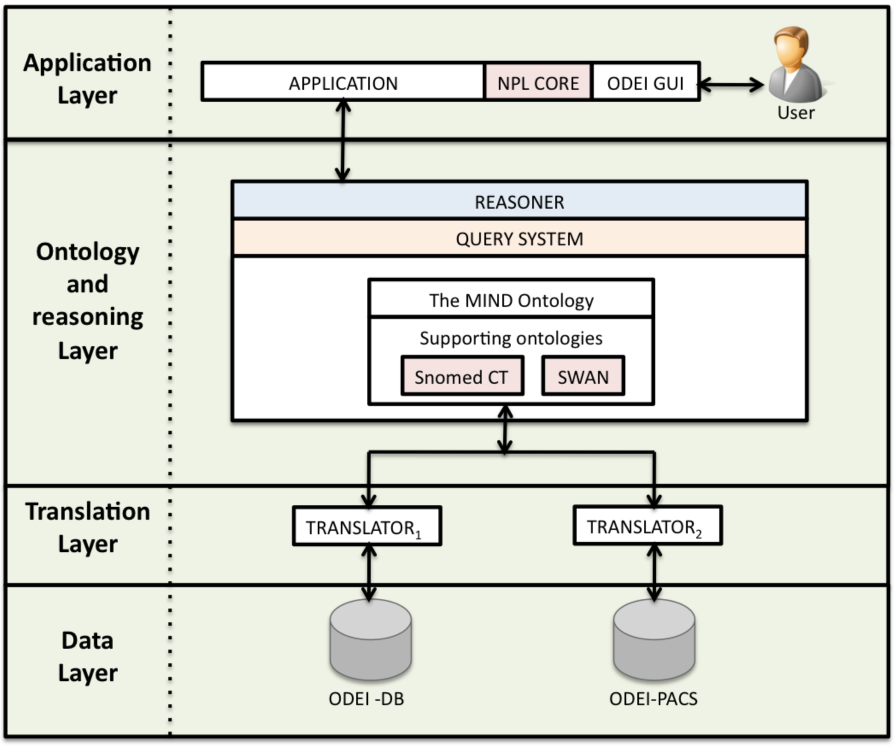

Figure 7.4: Architecture implementation for the CDSS of the MIND project

The **Data Layer** contains two DB: a MySQL database storing the results of the clinical tests carried out on the patients, called the ODEI Data Base, and a PACS, containing the DICOM image studies of the MIND Clinical Trial, called the ODEI PACS.

The **Translation Layer** is composed by XML data models, used to align the information stored in DB with the knowledge of the upper later. Every time new data is stored in the ODEI Data Base and the ODEI PACS, two XML documents are generated: (i) a xml schema document, containing the data structure, and (ii) a xml data document, containing the corresponding data. Such documents, are loaded into the ontologies of the upper layer, thus generating automatically the corresponding elements in the knowledge model and linking such elements to their data (query calls).

The **Ontology and Reasoning Layer** is formed by three modules:

The ontologies module: It contains the supporting ontologies SWAN, and SNOMED CT, and the domain ontology the MIND Ontology. The latter is implemented as an OWL DL ontology. Its loading and generation is performed using the Protégé-OWL API [5]. As an example of the ontology generation process: Figure 7.5 shows the data gathering process for the GUI of a part of the DB and 7.6 depicts the ontology generation for such part. Figure 7.7 depicts an example of a query-call instance.

[5] Protégé-OWL API web page (last accessed 31/01/2014): http://protege.stanford.edu/plugins/owl/api/

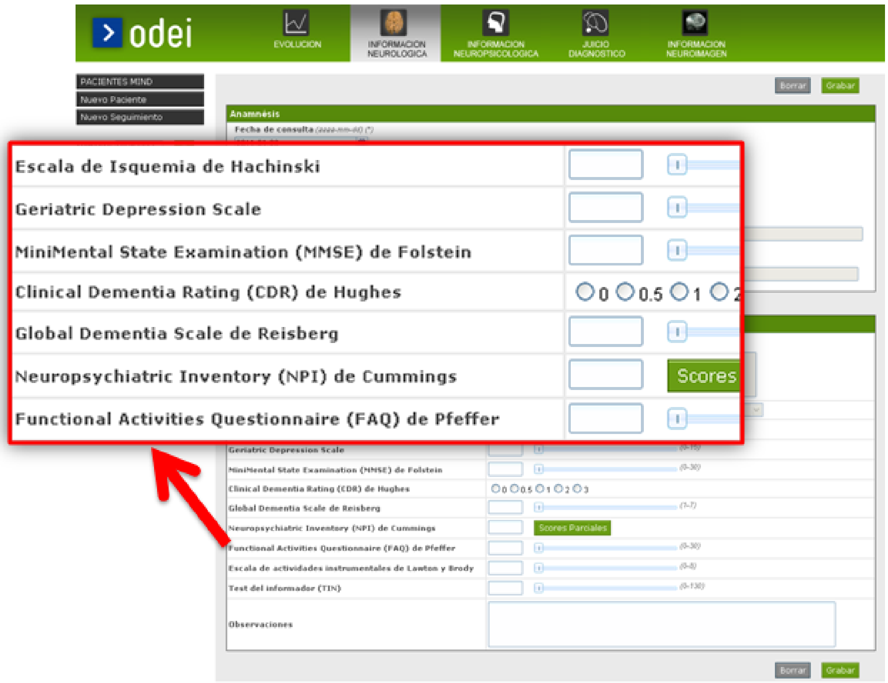

Figure 7.5: GUI for the data gathering process that is stored in ODEI Data Base

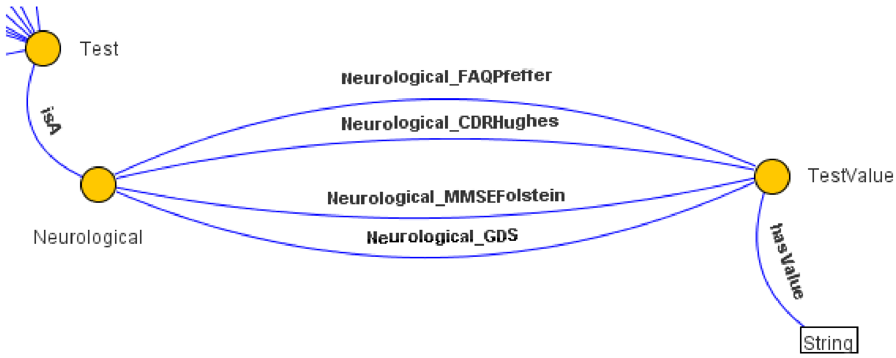

Figure 7.6: Ontology generation for the part corresponding to the data gathering in Figure 7.5

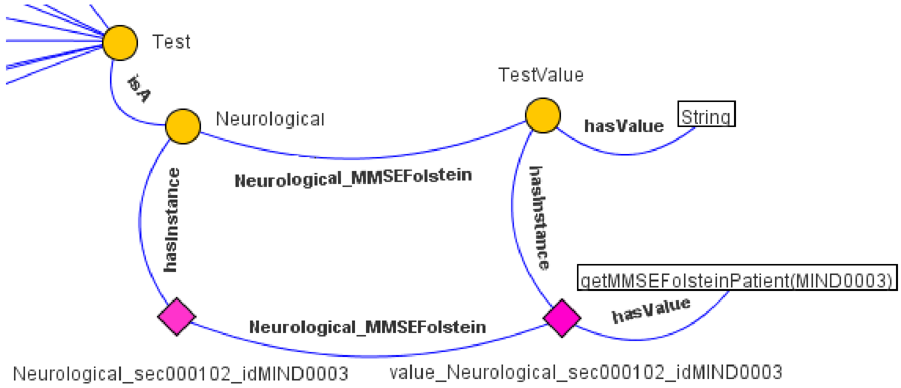

Figure 7.7: Instantiation to the query-call

The Query System: It implements the Reflexive Ontology technique for speeding up the querying process and providing automatic knowledge enrichment from the user queries (see Chapter 4 for specification and Chapter 8 for performance evaluation).

The Reasoner module: It performs a semantic reasoning process based on rules provided by domain experts, which model the process of AD diagnosis. The Reasoner applies the current set of rules in the system to the MIND Ontology inferring the corresponding diagnoses. For a given patient different rules may apply, each one carrying a corresponding diagnosis. The weights of the matching rules provide a ranking of the diagnoses for the final suggestion.

In the **Application Layer**, the ODEI Graphical User Interface (GUI) has been implemented, as well as a Natural Language Processing core, which translates the rules and their output, for the reasoner and the physician respectively. Figure 7.8 shows en example screenshot of the output recommendations provided to users, containing the inferred diagnoses.

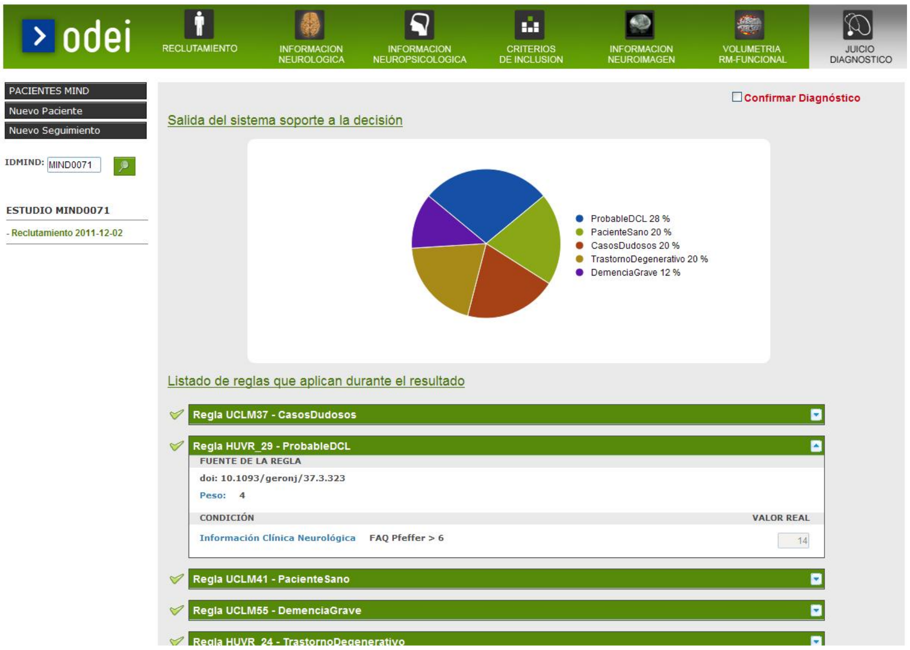

Figure 7.8: Example output provided to users containing the inferred diagnosed

7.6 Discussion - Towards the evolution of the ruleset for the early diagnosis of AD

In this chapter we have presented a Knowledge Engineering diagnosis-support tool for the detection of AD where ontologies and semantic reasoning play a fundamental role. Our work follows a knowledge-based approach, in which diagnoses are inferred by a rule-based Reasoner applied over ontologies.

However, in order to address other tasks such as knowledge discovery, the semantics supported by our knowledge based approach is not enough. In order to do so, the feedback of daily decisions of users is needed. That leads us to the next level, the experience handling level. In Chapter 9 we provide such implementation.

As part of the clinical trial for which this CDSS has been developed, it could be interesting that, for each patient included, the system could learn from the final decisions made by clinicians in comparison to the recommendations the system provided. In this way, if significant differences between recommendations and final decisions occur, the system could automatically improve the corresponding rules to provide better recommendations. In Chapter 5 we presented three different ways in which the ruleset could be improved: (i) generation of a new rule, (ii) deprecation of a rule, and (iii) fine-tuning of an existing rule. We have developed such system, but due to the requirements imposed by our industrial partners, the implementation was carried out under the research framework of the LIFE project on Breast Cancer, to be presented in Chapter 9. However, the results presented there can be directly translated to the domain of the AD, as a continuation of the work reported in this Chapter.

Chapter 8

Performance evaluation of Reflexive Ontologies for decision support

This Chapter presents a performance evaluation of Reflexive Ontologies (RO) carried out under the framework of the research project MIND (see Chapter 7). The main objective is to present a benchmarking that endorses the Reflexive Ontologies approach in a real scenario within the medical domain.

This Chapter is structured as follows: Section 8.1 introduces the technical challenge to overcome. Section 8.2 summarizes our contributions. Section 8.3 provides a short overview of the state-of-art on fast querying techniques. Section 8.4 describes implementation details. Section 8.5 details the evaluation experiment carried out to test the RO time response improvement. Finally, Section 8.6 discusses efficiency in terms of time provided by RO.

8.1 Description of the technical challenge

In the semantic domain, query processes provide both, users and computerized applications with mechanisms for interaction with ontologies and data [KGH11]. In the specific case of ontological querying, the result is the set of instances matching the query conditions (or assertions). As retrieval processes consume computer resources, systems that perform queries in a massive way would in general render in hungry resource demands. In the aforementioned scenario, fast querying systems are requiered in order to reduce the time consumption needed by querying tasks.

The RO preseve the updated map between queries and the instances matching such queries in the same knowledge model, so that they can speed up querying processes, when those queries are already presented to the system.

8.2 Brief description of our contributions

We have implemented RO in a real CDS environment evaluating its performance, in order to assess the efficiency improvements achieved (in terms of timing). Particularly, our implementation environment is the CDSS for early diagnosis of AD, that was developed within the framework of the MIND project presented in Chapter 7. In the MIND implementation of the CDS architecture (1^{st}generation) proposed in Chapter 6, we developed a semantic Reasoner to infer the corresponding diagnoses of a patient, according to the data associated to her/him.

8.3 Semantic and database queries

Relevant work has been reported concerning performance improvement of query answer procedures over knowledge bases. Kollia et al. introduced in [KGH11] optimization techniques that improve query answering performance for SPARQL-OWL queries. One of the optimizations presented in [KGH11] consists in utilizing precomputed information (e.g. the class hierarchy) in order to find the answer of a query by means of a cache lookup. This technique, along with some other optimizations, such as the axiom reordering, is proven to be able to achieve performance improvements in response time. Our approach is similar to [KGH11] in the sense that it benefits from previously computed information in order to perform a cache-like access to the query-answer vector. However, our approach goes further in the sense that RO keep track of all of the queries made over the ontology instead of using some pre-computation made by the reasoner. RO maintains an updated map of query-answers that guarantees a correct response regardless of data variations.

Amir et al. introduced in [Ami05] an approach known as partition-based logical reasoning to improve the efficiency of the reasoning process, giving algorithms for reasoning with partitions of related logical axioms. In [GPSK05], Grau et al. proposed the idea of partitioning an OWL ontology into sub-domains (modeled as separate ontologies) using e-Connections to combine them. This approach was intended to reduce the Knowledge Base portion that the reasoner has to work with, by keeping irrelevant components of the ontology unloaded. Both, [Ami05] and [GPSK05], are based on the idea of reducing the search-space within the knowledge base in order to improve reasoning efficiency. Our work tackles the reasoning time optimization issue from a different perspective as it is based on query caching rather than ontology partitioning.

The specific feature of RO that is responsible for the acceleration of the querying process is similar to the query caching techniques traditionally used in the context of relational databases. In particular, within the domain of web applications, many different techniques such as those in [ALK+02, APT03] have been presented in order to generate efficient caches from web content in constant transformation. Analogously, a RO has the ability to generate and maintain an efficient cache system even when dynamic knowledge bases are involved. RO have the ability to maintain a history of the queries made over the ontology itself, which acts as a cache thanks to the faculty of query retrieval [Tor08]. In addition, RO has the ability to provide a more refined cache system, other than a simple query history, thanks to the property of self-reasoning over the query set. In this manner the size of the cache can be reduced (or extended depending on specific needs) according to implicit knowledge.

Cobos et Al. proposed in [CTS+08] an architecture which uses the Reflexivity concept in order to perform a fast semantic retrieval in the Film Heritage domain. The results of the experiment showed a clear efficiency gain, with an improvement of two orders of magnitude in the execution time. Although the concept of using RO for a fast query recovery is the same for both cases, the architecture differs in some extent from our approach. The experiment by Cobos et Al. [CTS+08] was carried out using only simple queries (containing a simple condition clause) and the ontology they used within the experiment included 63 individuals. In this chapter we apply the Reflexive Ontologies concept in a more complex environment, since we use complex queries and our domain ontology contains more than 10^4 individuals. In addition, our system handles a non-static ontology (i.e. an ontology that grows over time), while the system by Cobos et Al. [CTS+08]works with a static ontology. The implementation of the Autopoiesis concept makes possible the use of non-static Reflexive Ontologies.

8.4 Implementation of the query system

RO has been implemented as part of the Clinical Decision Support System (CDSS) for the early diagnosis of AD developed within the MIND project, presented in Chapter 7. Rule antecedents are implemented as query clauses q_i and launched at the ontology to get the set of individuals I_{Q_i} that match conditions. Then rule consequents are applied to I_{Q_i}.

In Chapter 4 we specified a simple query clause by a tuple $q_i^s = \langle V_i, m_i, v_i \rangle$, where V_i is a variable, m_i is the comparison operator (i.e. $>, <, =$) and v_i a value of the range of V_i. In our implementation, variables V_i correspond to Object type properties P_i^o whose domain is class C_j. Let us define a relating operator $\mathcal{C}\left(q_i^s, C_j\right)$ that is true if V_i of q_i^s corresponds to a property P_i^o whose domain class is C_j.

On the other side, a complex query clause q_i^c was specified by n simple queries, combined by logical operators, θ, (i.e. $\vee$, $\wedge$ and $\neg$) that define the relationships among consecutive simple queries:

$$q_i^c = \{(\theta_n, q_n^s)\}_{\forall n},$$

where θ_n is the n-th logical operator (i.e. $\vee$, $\wedge$ and $\neg$), and for consistency we assume that $\theta_0 = \emptyset$.

When a query clause q_i is launched at the ontology, the RO module searches through reflexivity instances in order to check whether it has been made previously. If the system finds a reflexive instance for q_i , the answer I_{Q_i} is retrieved directly from that instance. When the query is not present in the RO query pool, a traditional query is made via a Java compliant API to generate I_{Q_i}.

If query q_i is complex q_i^c and not present within reflexivity instances, it is first split into n simple query clauses q_n^s by a query parser. Each simple query q_n^s is then searched through reflexivity instances and if it has not been previously queried the corresponding matching individuals are generated in the traditional way. Matching individuals for each simple query I_{Q_n} are then combined by the corresponding logical operators θ_n, in order to obtain the resulting matching individual set I_{Q_i} for the original complex query q_i^c.

For every new query matching instances are calculated in the traditional way (both complex and simple), the reflexivity instance $Q_i = (q_i, I_{Q_i})$ is added to the RO structure. Figure 8.1 illustrates the query process in a RO-based system.

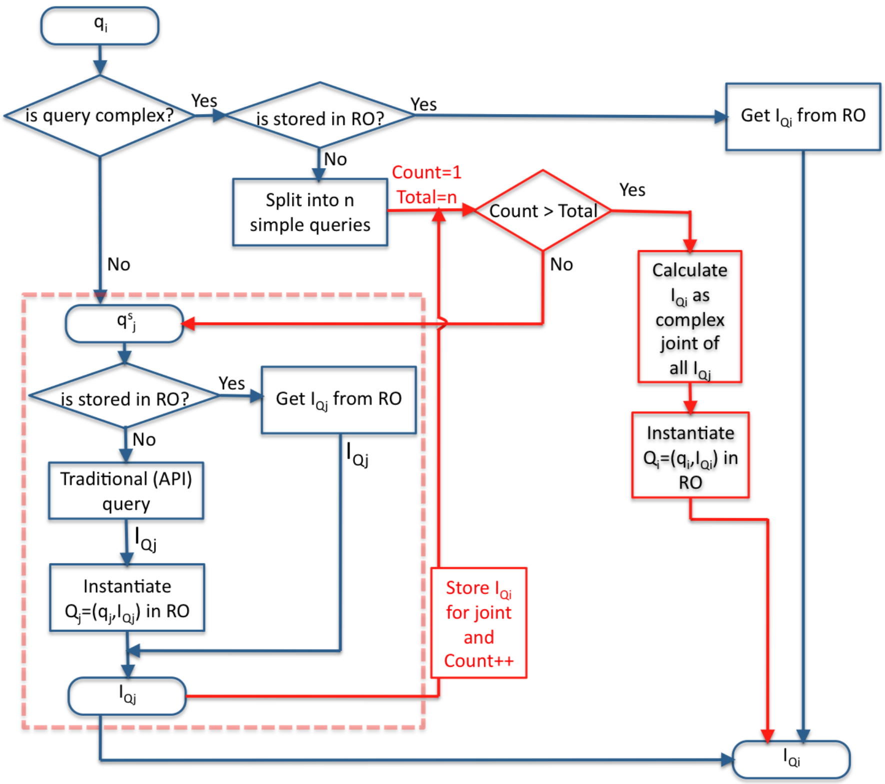

Figure 8.1: Flow of the query process in Reflexive Ontologies

8.4.1 Integrity update implementation

As already discussed in Chapter 4 the integrity update property of RO guarantees that any change in the ontology may be reflected in the reflexive structure in order to maintain data consistency. When an external perturbation takes place (for instance, the modification of an individual of the ontology), the system editor ε can act in different ways. Hence, during the design and implementation process of the system, different paradigms have been considered:

Query instance removal: The first paradigm is based on the detection of modified instances, removing from the RO queries related with the class of the modified instance. This approach removes the whole list of related individuals that the RO query contains, that means that a single individual would penalize the entire set. In return, the computational cost of this method is relatively low.

Query instance update: The second paradigm is based on updating the list of RO queries, modifying the list of related individuals of the corresponding query. Algorithm 8.1 shows the pseudocode of the implemented RO integrity algorithm. This approach has a relatively high computational cost since it requires performing pre-caching work; therefore its use is only recommended in application domains where data has a limited variability.

8.4.2 Autopoiesis implementation

The autopoietic property of a RO refers to its ability to self-creation or self-organization after being subject to external perturbations as well as to implicit knowledge coming from previous queries (see Chapter 4). It is tightly related to the integrity update property, but we also implemented a method to modify the query set depending on the use that have been made of the various classes of the ontology. Based on the assumption that queries are not made at random, so that they follow patterns, we have designed a simple system to measure the use of each class assigning a factor accordingly.

Let $\mathcal{Q}^T$ be the set of simple query clauses in our reflexive structure, and $\mathcal{Q}_j$ the set of simple query clauses that refer to a class C_j, such that

$$\mathcal{Q}_j = \{q_i^s \mid \mathcal{C}\left(q_i^s, C_j\right)\}$$

An Occurrence Factor F_j is assigned to each class C_j depending on the number

Algorithm 8.1 Pseudocode for the Reflexive Ontology Integrity implementation algorithm

```
Pre: The modified individual is known I_i
Post: updated RO query set
function updateROQueries(I_i)
{
    C_i= getClassFromIndividual(I_i)
    for ρ=1 to Number of Query Instances in RO
    {
        V_ρ=getVariableFromQuery(q_ρ)
        C_ρ=getDomainClassRelatedToVariableProperty(V_ρ)
        if C_i=C_ρ then
        {
            for j=1 to Number of individuals in I_{Q_ρ}
            {
                if I_j equals I_i then
                {
                    Remove I_j from I_{Q_ρ}
                }
                if M(q_ρ, I_i) then
                {
                    Add I_i to I_{Q_ρ}
                }
            }
        }
    }
}
```

Algorithm 8.2 Pseudocode for the Reflexive Ontology Autopoiesis implementation algorithm

Set threshold T_h
$\mathcal{Q}^T$=0
for j=1 to Number of Classes in Ontology **then**
{
    $\mathcal{Q}_j$=0
    Initialize Q_j
}
for ρ=1 to Number of Query Instances in RO
{
    get C_j related to q_ρ
    $\mathcal{Q}_j$++
    Add q_ρ to Q_j
    $\mathcal{Q}^T$++
}
for j=1 to Number of Classes in Ontology **then**
{
    $F_j=\frac{\mathcal{Q}_j}{\mathcal{Q}^T}$
    if $F_j<T_h$ **then**
    {
        Perform action (delete, modify, etc) to queries in Q_j
    }
}

of reflexive queries referring to that class, such that

$$F_j = \frac{|\mathcal{Q}_j|}{|\mathcal{Q}^T|}$$

Once the Ocurrence Factors for each class have been calculated, a threshold value could be arbitrarily established. Queries q_i^s contained in Q_j and assigned with a factor F_j below the threshold could be removed. If q_i^s are part of a complex query clause q_i^c, the complex query would only be removed if all q_i^s that conform them are marked as removable.

Algorithm 8.2 shows the pseudocode of the implemented RO autopoiesis algorithm.

Table 8.1: Size and characteristics of the rule sets

Rule set	Number of Rules	Number of Queries	Generating query instances
RuleSet1	35	120	72
RuleSet2	103	339	246
RuleSet3	138	459	291

8.5 Reasoning and query system: evaluation of time efficiency in the CDSS

We have carried out a benchmarking experiment in order to assess the performance differences between systems using RO and conventional ontologies. Simultaneously, the MIND system has been used to design and test new techniques that delve into the extraction of implicit knowledge from the RO. The goal of this comparative evaluation is to determine the reliability of the system and the effectiveness of the proposed optimizations, in a quasi-real life environment.

Methodology

For the performance evaluation of the system the execution time of the diagnosis process was measured against an arbitrary number of 10 patients. In order to compare the differences in performance, the system was implemented in two different ways: i) using RO, extending the MIND Ontology presented in Chapter 7 with reflexivity, and ii) without using them (no-RO), querying the MIND Ontology in the traditional way.

The system was implemented using the three different rule sets as shown in Table 8.1. For each rule set the number of rules is shown, along with the number of queries, and the number of corresponding RO queries that are generated inside the reflexive structure. The number of queries depends on the number of simple queries contained by each rule. The number of generated query instances depends on the number of queries that are repeated. For testing purposes we have randomly selected 10 patients which are subjected to diagnosis in each of the system configurations.

Execution time is retrieved following a methodology that is similar to the one used in Knowledge Base System benchmarking [GPH04, BHJV08, MYQ+06]. However, our evaluation process differs from these approaches in the fact that our measures are not taken from the execution of individual queries, but from the entire diagnosis process. This fact, however, does not detract validity to our evaluation, since the diagnosis process itself can be regarded as a sequence of queries. Execution time is measured using built-in Java methods with every measurement being the average of 10 independent executions.

Computing environment

We have performed the experiment in a desktop computer with Intel Core 2 Quad CPU Q8300 at 2.5 GHz x 4, 2.9 GB RAM and Ubuntu 11.10 64-bit. The test system was implemented in Eclipse 3.7.0 with JDK version 1.6.0 and Protégé-OWL API version 3.4.2 was used for ontology access and management[1].

Data and analysis

For each of the 10 patients, the time expended on the diagnostic process is measured in both approaches: RO and no-RO. Table 8.2 presents a complete report of the test results (execution time is given in milliseconds). In both approaches, the three rule sets shown in Table 8.1 were used. Table 8.2 shows that, using the same rule set, execution times vary slightly from one patient to the other. This occurs with all the sets of rules in both system configurations, i.e. RO and no-RO. This is caused by the fact that the number of queries to be performed at diagnosis time is determined by the complexity of each rule in the rule set. The rule set remains constant for every patient, thus the queries to perform are equal for the whole group of patients. A preliminary analysis suggests that small variations are caused due to differences in the number of clinical test instances between patients. Aside from those differences, the reduction of the execution time needed to perform the diagnosis by the RO system is evident. It is better shown in Figure 8.2, where the average execution times are compared for each of the rule sets. These values are

[1] Protégé-OWL API web page (last accessed 31/01/2014): http://protege.stanford.edu/plugins/owl/api/

Table 8.2: Execution times in milliseconds for each rule set and patients using both RO and no-RO systems.

Patient	RO / no-RO	RuleSet 1	RuleSet 2	RuleSet 3
1	RO	825	1813	2541
1	no-RO	2715	4139	6343
2	RO	809	1888	2522
2	no-RO	2688	4107	6329
3	RO	771	1882	2488
3	no-RO	2788	4100	6290
4	RO	858	1923	2538
4	no-RO	2752	4165	6298
5	RO	972	1932	2553
5	no-RO	2729	4207	6281
6	RO	772	1869	2637
6	no-RO	2705	4166	6229
7	RO	827	1831	2615
7	no-RO	2726	4097	6238
8	RO	878	1989	2524
8	no-RO	2720	4130	6241
9	RO	767	1976	2436
9	no-RO	2781	4181	6240
10	RO	785	1886	2537
10	no-RO	2802	4213	6146

obtained by calculating the arithmetic mean of the execution times measured for the 10 patients under consideration. The results show that the use of RO significantly reduces execution times (69.8% for RuleSet1, 54.2% for RuleSet2 and 59.4% for RuleSet3).

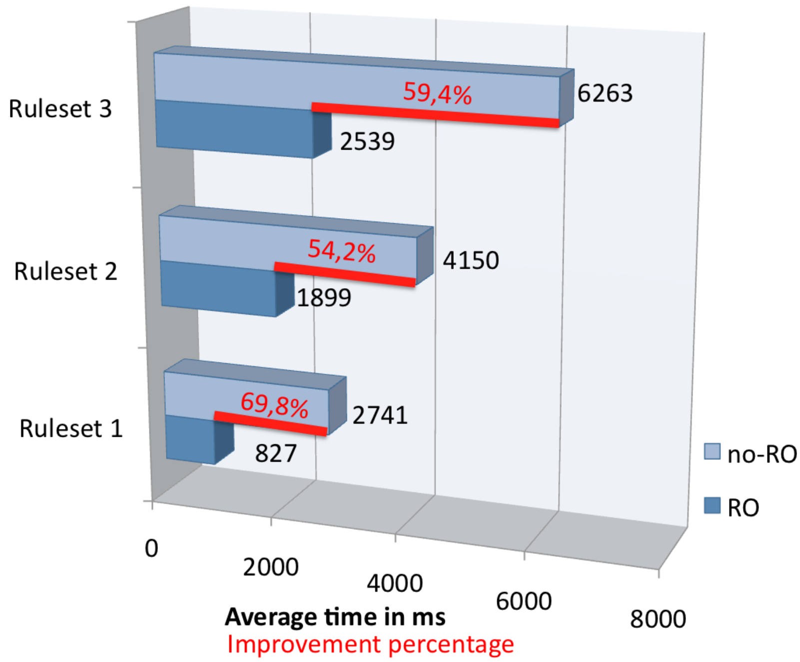

Figure 8.2: Average execution times

Comparing the execution times of both approaches in relation to the number of queries may bring valuable information. Figure 8.3 shows the evolution of the execution times as the number of queries to be performed increases. The number of queries contained in the set of rules has a major impact on the performance of both systems, since this number determines the time required to perform the whole diagnostic process. The graph in Figure 8.3 shows that the execution times for RO grow linearly. However, the execution times for the conventional ontology grow, if not exponentially[2], at least counterpart. This means that, compared to a conventional ontology, a RO is more robust to variations in scale of the system. As a conclusion RO improves the scalability, since the execution time growth is significantly slower in relation to the number of queries.

[2]Three points are not enough to accurately extrapolate this data

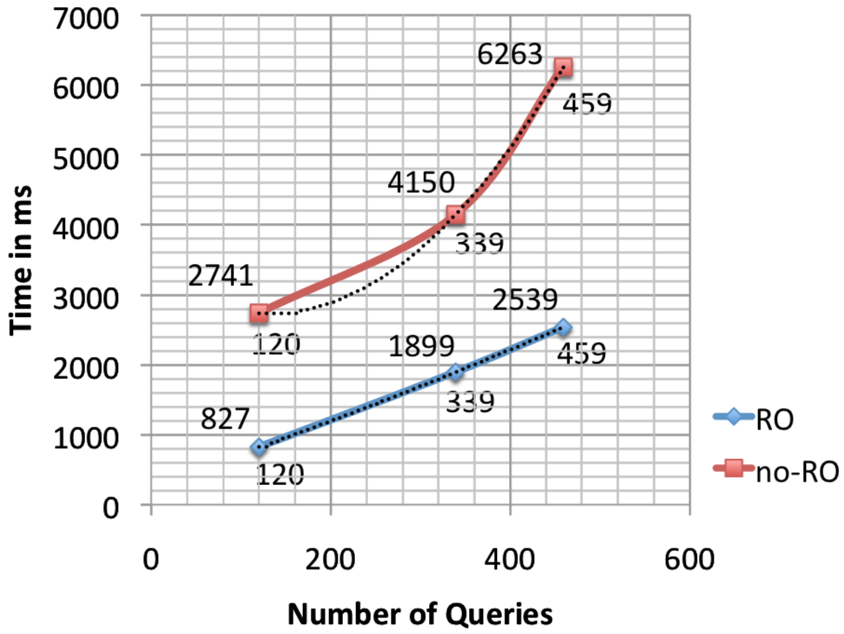

Figure 8.3: Execution time in relation to the number of the queries in the rule set

Figure 8.4 shows the evolution of the execution times in relation to the number of rules in the rule set. Although the trend is similar to the one presented in Figure 8.3, the number of rules may not be as significant as the number of query clauses they contain, since the complexity of the rules (in terms of number of clauses) may vary widely. This interesting fact means that, for instance, a rule containing ten clauses consumes the same computing time as a rule containing only two, when the computational cost of their processing is clearly uneven. Both charts show that the reduction of the execution time is fairly pronounced when reflexivity is used. Growth difference between the two smallest sets of rules is similar, even if the growth of the conventional ontology is about 40% greater than the RO. Nevertheless, the difference is more evident when the biggest rule set is involved. Figures 8.3 and 8.4 show an increase of more than 2100 milliseconds when the size of the rule set grows 35 rules (containing 120 queries). Under the same conditions, the execution time grows 640 milliseconds if the ontology is enhanced with reflexivity that is, 69.7% less.

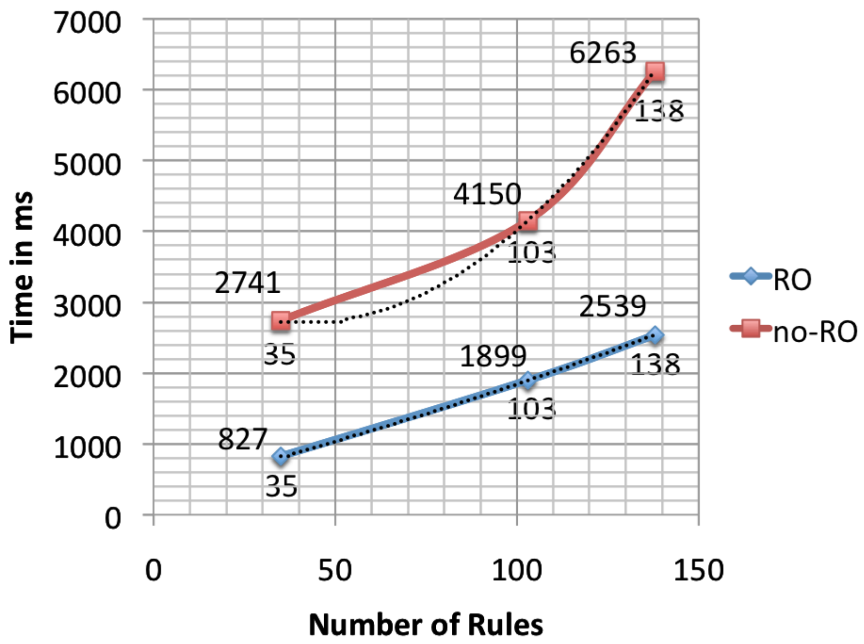

Figure 8.4: Execution time in relation to the number of rules in the rule set

From the obtained results, in addition to the exposed conclusions, further information can be extracted. As presented above, the analysis of RO queries provides valuable information such as the Occurrence Factor $\mathcal{O}$ of the classes. Figure 8.5 shows the graph of the MIND ontology, where both the size and color nodes refer to the $\mathcal{O}$ of a class. Taking into account that $\mathcal{O}$ is directly related to the number of times that a class appears in a query, the bigger and darker a node is, the more often it has been queried. According to that, Figure 8.5 shows that the class representing Neuropsychological tests is the most frequently queried, followed by the Neurological tests class. This could lead to interesting second order analysis of a qualitative nature meaning, for the example at hand, that the domain experts were strongly biased towards Neurology.

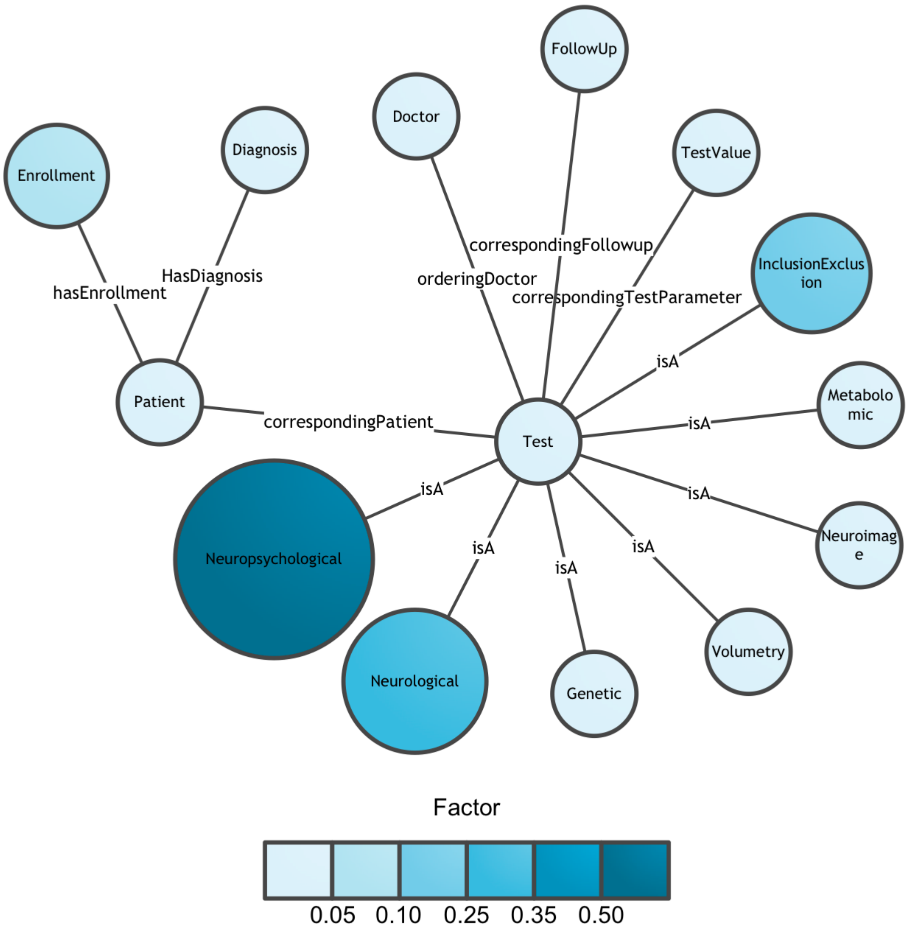

Figure 8.5: Graph representing MIND ontology

8.6 Discussion - Temporal efficiency of RO

The implementation of the Reflexive Ontologies (RO) in a knowledge-based CDSS for the diagnosis of Alzheimer's disease has been tested to measure the impact of RO in terms of temporal efficiency gain. We have presented a benchmark experiment that compares two systems: one of them using RO, and the other using a conventional ontology. Our comparative evaluation suggests that, in the worst case scenario, RO perform comparably to traditional ontologies. It also shows that in our application scenario (a diagnosis system) the use of RO significantly improves efficiency, reducing the execution time by almost 70% at the best case, and by almost 55% at the worst.

However, the application of RO in other scenarios may not result on perfomance improvements as high as ours. The appropriateness of scenarios depend on the following criteria:

(i) the complexity of the ontology:more complex ontologies have grater search time to answer a query, so that the RO reduction in search time produces a more noticeable benefit,

(ii) the number of individuals of the ontology, again the higher the number of individuals in the ontology the grater the search time that is saved by the RO prefetched query answers,

(iii) the number of rules: the higher number of rules the lower benefit because maintaining the integrity of each query requires the reevaluation of all rules that are related to it,

(iv) the number of queries per rule: the higher number of queries per rule decreases the benefit of RO because the cost of preserving the integrity of the cached queries increases accordingly, and

(v) the query variability between different rules: the increasing similarity between queries of different rules imply that the actual collection of clauses to be evaluated is smaller, so that RO increases its benefits..

The limit of RO performance improvement is theoretically achieved when the time required for calculation of the matching instances I_{Q_i} of a query clause q_i in the traditional way equals the time consumed while searching over the reflexive structure (array). Huge reflexive structures will thus deteriorate the system performance.

Additionaly, the process of integrity updating generates an extra workload for a system using RO. Every time a new individual is added to the ontology, or an existing individual is edited or deleted, the queries matched by such instance need to be updated. Such workload must be compensated by the speeding up achieved during instance matching, but it needs to be performed at instance edition time. In the MIND scenario, edition and matching tasks are temporally distant, due to the peculiarities of the clinical trial, where patients must go through all the different tests (i.e. neuropsychological, neurological, radiological, metabolomical and genetic) before returning to the neurologist, who performs the diagnosis. Each medical appointment with a different speciality professional generally takes place at different dates. Therefore, the system integrity update workload occurs at different time frames than the workload imposed for diagnosing (instance matching). Nevertheless, other clinical scenarios, such as daily practice may not show the same benefits as in MIND, owing to the fact that edition and matching often will be performed by the same professional and at the same time.

In this regard, the adaptation of the use of RO depending on the scenario is still an open research issue. It will be highly desirable to be able to predict the adequacy of RO in terms of the above criteria, that is, to have some kind of suitability measure that would allow to activate or preempt the RO. Even better would be to activate RO selectively to specific queries and rules, so that some parts of the system would benefit from the speed up, avoiding the excessive workload of integrity preserving in other parts of the system.

As an improvement of RO we propose in Chapter 4 the Extended Reflexive Ontologies (ROX), including not only queries into the model, but also rules. Further tests measuring the efficiency provided by ROX are still needed.

Chapter 9

S-CDSS for Breast cancer

This Chapter presents some of our technical contributions in the scope of an applied research project dealing with the diagnosis, treatment and follow-up of Breast Cancer, called Life. Our contribution to the project is based on some of the already presented techniques in chapter 5.

This Chapter is structured as follows: Section 9.1 introduces the technical problem and identifies challenges to overcome. Section 9.2 summarizes our contributions. Section 9.3 provides a short overview of the state-of-art on current CDSS on Breast Cancer. Section 9.4 proposes an architecture for a S-CDSS for diagnosis and treatment of Breast Cancer, as well as implementation details. Section 9.6 describes the evaluation methodology of such system. Finally, Section 9.7 sumarizes the obtained conclusions.

9.1 Description of the technical problem and challenges identified

Regardless of race or etnic group, Breast Cancer is the second most frequent cancer in women (being skin cancer the first). According to the US Centre of Disease and Prevention [Gro13], in 2010, 206.966 women were diagnosed with breast cancer in the US, and 40.996 died from the disease.

The detection of the disease at early stages allows the use of more effective and less invassive treatments, providing better recovery results. For that purpose preventive screening programs are performed by medical institutions to women in the age interval of higher incidence of Breast Cancer, after accumulated evidence proving their effectiveness on early detection. In this direction, mammography screening has been extended, doubling the number of early diagnoses achieved [BW12]. Other techniques, such as ultrasound imaging, are applied along with mammography, providing better detection results in some cases, for instance in dense breasts [JMM+13]. In particular, in Spain, annual mammographies are perfomed to women in ages ranging between 50 and 70 years-old[1].

From a medical research perspective, there is still a need of finding evidence of the cost-effectiveness of preventive actions on other collectives of women in risk. The aforesaid evidence could help to drastically reduce the cases when diagnosis is done at high stages. In order to do so, clinical trials are needed to identify relevant inclusion criteria on screening programs. Published research [NGM14, HD03] has identified some of the urgent challenges that need to be taken into account, being the following the most notorious and relevant to our investigation:

Challenge 1: Personalized therapies At a molecular level many different types of breast cancer have been identified. Each type requires specificic treatments to be prescribed patient-wise. It is the common understanding that a treatment that could eventually work in one case, may be deadly in some other. With that in mind, it is essential that patient-specific characteristics are be taken into account in order to fully provide a personalized therapy. From the technical point of view, such a personalization requires that the system is able to (i) characterize the patient and its profile, and (ii) recommend the most appropiate therapy for each patient.

Challenge 2: Knowledge handling The process of generating decision recommendations involves the handling of both, (i) large amounts of patient information and (ii) the existing domain knowledge on the disease diagnosis and treatment. The latter is comprised by knowledge coming from bibliography (i.e. experience reported

[1]Donostia University Hospital, Breast Cancer Treatment Protocol (*in Spanish*), 2011 (report numer 48). Last accessed: 2013/12/04. URL: `goo.gl/eGk9Bx`

from other medical groups) and the knowledge generated from the personal experience of the physicians handling the case, or their groups.

Challenge 3: Knowledge updating and maintenance Both of the aforementioned sources of domain knowledge (bibliography and experience) are in continuous evolution. Thus, continuous updating of the knowledge in the system to keep pace with the evolution of domain knowledge sources is necessary. Classical approaches follow a manual methodology, where a domain expert needs to find the updated knowledge and introduce it in the system. Currently, tools to handle automatically their update and maintenance are desirable.

9.2 Brief description of our contributions

Considering the previously defined challenges, we developed a Semantic steered Clinical Decision Support System (S-CDSS) for diagnosis and treatment of Breast Cancer. The architecture of our system followed the 3rd generation CDSS architecture presented in Chapter 6 .

In brief our contributions can be summarized as:

- We propose the development of a S-CDSS, based on the experience handling methodology presented in Chapter 5. Its main objective is to enhance current knowledge with daily experience, in order to continously improve the provided decision recommendations.
- We proposed a methodology to generate the CDSS ruleset.
- We developed three different algorithms to evolve rule weights, adapt rule variables and generate new rules, depending on previous experience acquired by the system.
- We proposed an evaluation methodology for the system.

9.3 Brief review of current CDSS for diagnosis and treatment of Breast Cancer

During the last years, several Clinical Decision Support Systems for Breast Cancer diagnosis and treatment have been presented in the literature [JMM+13, CSJ+10, CHW06, DBF+13, SP09]. Most of them propose image-based analysis for optimizing diagnosis of Breast Cancer [JMM+13, CHW06, DBF+13, CSJ+10]. Mammography and ultrasound imaging are the most used techniques. Approaches covering the whole clinical process of Breast Cancer are also proposed, such as the CREDO project [SP09], where a PROforma based clinical guideline is implemented to aid doctors and patients along the different phases followed during diagnosis and treatment. In [AAHS07] a CDSS for Breast Cancer follow-up interventions at Primary Care Setting is presented. Their approach is similar to ours, in the sense that the recommendation generation process is similar: it is based on rules and on a JENA inference engine. They also provide a rule authoring module, based on the Guideline Element Model (GEM). Another example is the work in [CDV08], presenting KON, an Oncology CDSS based on knowledge management. It is specially focused on an approach integrated within the clinical system of the hospital and the clinical workflow, and an ontology and a set of rules are provided. However, no automatic knowledge maintenance is provided in the aforementioned works, highlighting the relevance of our approach.

9.4 Scenario for Breast Cancer diagnosis and treatment

Our work is currently being implemented in the Breast Unit (BU) of the Valencia University General Hospital. The BU is a multidisciplinary team of physicians,. In the aforesaid environment, decisions about the healthcare process of patients are made both individually and collectively. Their approach is to analyze and discuss the most critical cases in a weekly plenary meeting in order to reach common ground on the relevant aspects of their diagnosis and treatment. In the scenario described, potentially risky decisions depend on the combination of the knowledge and experiences of different professionals. In particular, the medical team in the LIFE project is conformed by the following hospital services:

(i) Radiodiagnosis, which is a service specialized on the use of imaging technologies, such as X-ray radiography, Ultrasound, Mammography and Magnetic Resonance Imaging (MRI), helping to diagnose diseases by visualization of the inner human body;

(ii) Nuclear medicine, a service specialized in the application of radioactive substances for the diagnosis and treatment of diseases;

(iii) Pathologic Anatomy, a service specialized in the diagnosis of a disease based on the gross, microscopic, chemical, immunologic and molecular examination of organs and tissues;

(iv) General Surgery, a service specialized in the surgery of the abdominal content, as well as in skin, breast, soft tissue, and hernias; particularly, breast tumor removal specialists are involved in the Life project;

(v) Medical Oncology, which is a service specialized in the treatment of tumors, primarily with drugs (e.g. chemotherapy and hormonotherapy);

(vi) Radiation Oncology, which is a service specialized in the treatment of tumors with radiation (i.e. radiotherapy);

(vii) Rehabilitation, a service specialized in the rehabilitation of physical injuries, that could also have been generated as a result of the surgery or another treatment; and

(viii) Psychology, a service specialized in supporting mental and emotional aspects of the disease.

Let us review a common clinical scenario starting when a patient is redirected to the Breast Unit after having discovered by palpation or by imaging (e.g. Breast Cancer screening) a lump in the breast. The patient will visit a series of specialists from different services. During those visits, doctors will perform the different tasks (i.e. diagnosis, prognosis, treatment, follow-up, prevention).

During the first visit, the clinician (usually a radiologist or a surgeon) starts with the diagnosis phase. In this stage, the gathering of the outmost relevant clinical history of the patient is secured, as well as the results of physical and complementary explorations when needed (e.g. medical imaging or pathological) It could happen that for the complementary explorations, the patient would be derived to other specialists (Radiodiagnosis, Pathologic Anatomy, Nuclear Medicine).

At ths point, the sumarized patient data is analyzed by the BU (during their weekly meeting) and an initial diagnosis is agreeded, as well as the prognosis and the treatment plans, based on their knowledge and prior experience. If diagnosis is not clear, more tests could be requested.

Depending on the treatment the patient could be derived to different services, such as General Surgery, Medical Oncology, Radiation Oncology, Rehabilitation and Psychology, which will also perform patient follow-up plans. It could happen that during the follow-up of the patient, some new symptoms reveal a variation on the diagnosis, for which a different treatment procedure would be required. Finally, after the tumor is removed, preventive follow-up visits will be scheduled.

During the whole process the BU members follow clinical guidelines on Breast Cancer. Assuming that they have an available S-CDSS, the different decisions involved during the process (both individual and collective) would be supported and stored, representing a valuable asset. In order to support such tasks, our system must provide integration at the levels of (a) clinical data and results from the different services, (b) domain knowledge and criteria for the decisions involved in each of the services, and (c) the experience acquired during the individual and collective decision making process.

9.5 System implementation

In order to provide a computerized approach for the described problem, we have implemented the 3rdgeneration architecture presented in Chapter 6. As agent implementation details are provided in Chapter 6, this Section focuses on the description of our implementation of the data repository, the knowledge repository and the experience repository.

Implementation of the data repository

At data level, a mySQL database has been implemented for the unification of the eight different databases present in the hospital (a database per medical service). We call this database LifeDB and it contains the information needed by the team of physicians and our system to make decisions. The process is implemented by, (i) analyzing the current databases, (ii) gathering the requirements and needs of each of the services that should be covered by the new design, and (iii) then aligning such databases. For the requirement gathering phase, a large spreadsheet document, called the LIFE Data Model, has been generated and is iteratively validated with the medical team, every now and then. All recorded patient data and variables involved during decision-making processes have been added to such document, as well as their types and ranges.

Implementation of the ontologies in the knowledge repository

The data structure of LifeDB is aligned with the KREG model in the knowledge repository. To do so, we have implemented a Translation Agent, which creates two xml documents in real time every time data is created or modified in LifeDB: (i) one xml document contains the data structure and (ii) the other has the query calls to the LifeDB. These two xml documents are programmatically loaded to the Knowledge Layer of the KREG model using and API call from the Protégé-OWL API [2].

In our implementation of the Knowledge Layer, we have followed the methodology proposed in Chapter 6, which consists on the mapping of three different ontologies: (i) an upper standard ontology for the clinical domain, (ii) an ontology for the representation of bibliography, and (iii) a domain ontology for the diagnosis,

[2]Protégé-OWL API web page: `goo.gl/NmGirH`

treatment and follow-up of Breast Cancer. For that purpose, we have mapped the following three ontologies:

(i) SNOMED CT [NVN+10], for clinical description of the patient, the breast cancer and the procedures involved during its diagnosis and treatment;

(ii) SWAN [CWW+08], for bibliographic endorsement of criteria for decision making, and

(iii) a new domain ontology of Breast Cancer, containing the results of the specific clinical tests carried out to patients that we name the Life Ontology (a partial view of the Life Ontology is depicted in Figure 9.1).

Three main classes form the Life Ontology: **Patient**, **Doctor** and **EHR**. **EHR** stands for Electronic Health Record and contains all patient-related general, sociological and clinical information. These three types of information are reflected in the three subclasses of **EHR**:

i) **General_Information**,

ii) **Socio_Demographic_Information**, and

iii) **Medical_Tests**.

Two main object type properties relate the three main classes: *correspondingPatient*, linking an **EHR** instance with a **Patient**, and *orderingDoctor*, linking an **EHR** instance with a **Doctor**. Subclass **Medical_Tests** contains eight different subclasses, one for each service of the BU: **Radiodiagnosis**, **Nuclear_Medicine**, **Radiotherapy**, **Rehabilitation**, **Anatomical_Pathology**, **General_Surgery**, **Medical_Oncology**, and **Psychology**.

The variables contained in the LIFE Data Model are reflected in the LIFE Ontology, by means of data type properties whose domains are these eight classes. An example is depicted in Figure 9.1, where two data type properties related to the **Radioagnosis** class (service) are shown:

i) the BIRADS [oR13] value for the mammography (*RD_Mammography_BIRADS*), and

ii) the BIRADS [oR13] value for the ecography (*RD_Ecography_BIRADS*).

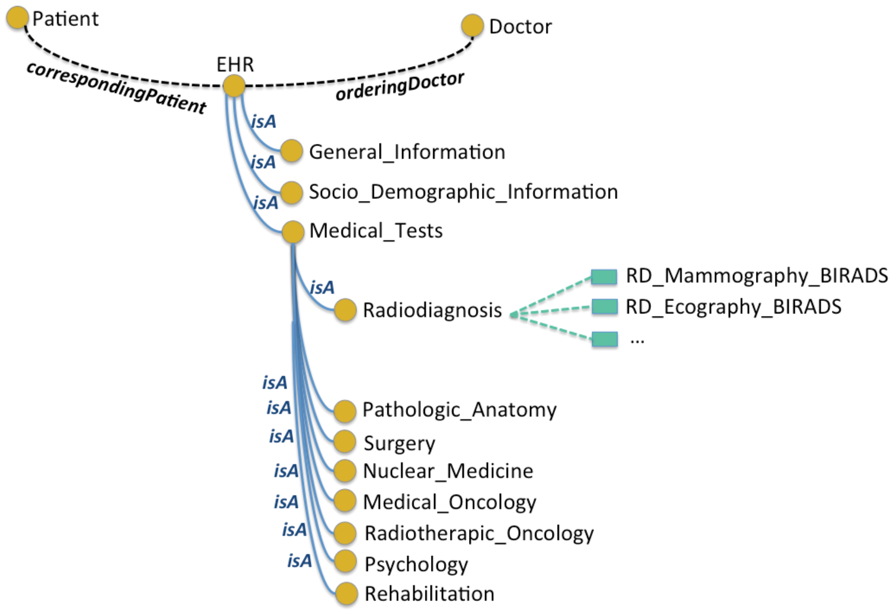

Figure 9.1: The Life Ontology

The knowledge and decision agent implements tools and techniques oriented to medical domain experts, for the edition and visualization of these domain ontologies. In particular, a graph-based ontology mapping tool developed by other members of our reaserch team [Art11] has been used for this purpose, as shown in Figure 9.2.

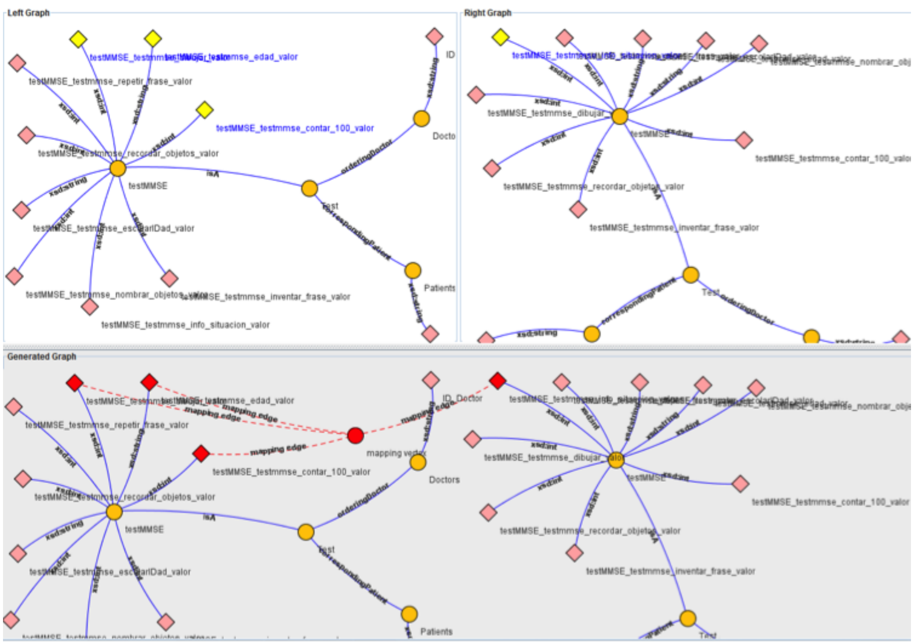

Figure 9.2: Screenshot of the ontology maping and visualization tool

Implementation of the production rules in the knowledge repository

The Rule Layer of the KREG Model consists of an initial set of production rules generated by the doctors of the BU. Those rules model the different localized decisions for each service. As there was no previous experience on rule formalization in the hospital, the medical team has proposed a rule generation methodology that we have implemented in order to ease and speed-up the rules-gathering process.

1. First, for each variable included in the LIFE Data Model (e.g. radiotherapic protocol type, from the radiotherapic oncology service: class **Radiotherapy**, property *RT_ProtocolName*), physicians identify whether it depends on other variables. For instance, the type of radiotherapic protocol applied to each patient depends on

   (a) the type of surgery applied
   (class **GeneralSurgery** property *GS_InterventionType*),

   (b) the size of the surgical piece of pathologic anatomy
   (class **PathocologicAnatomy** property *PA_SurgicalPiece_Size*),

   (c) the number of lymph nodes
   (class **PathocologicAnatomy** property *PA_SurgicalPiece_*LimphNodes), and

   (d) the existence of hypersensibility
   (class **Radiotherapy** property *RT_Hypersensibility*).

2. Then, the dependence conditions for every different possible value of the former variable are established (e.g. radiotherapic protocol MAMA-50 is recommended when the type of surgery applied is conservative, there are no lymph nodes, there exists hypersensibility, and the size of the surgical piece of pathologic anatomy is T0, T1mic, T1a or T1b).

3. Finally, rules are generated from the aforesaid conditions and introduced to a web based rule generator tool that was generated by other members of the

research team and that has been integrated within the S-CDSS application (knowledge and decision agent implementation). It allows to easily create rules, without dealing with rule syntax. Figure 9.3 shows a screenshot of the rule generation graphical tool.

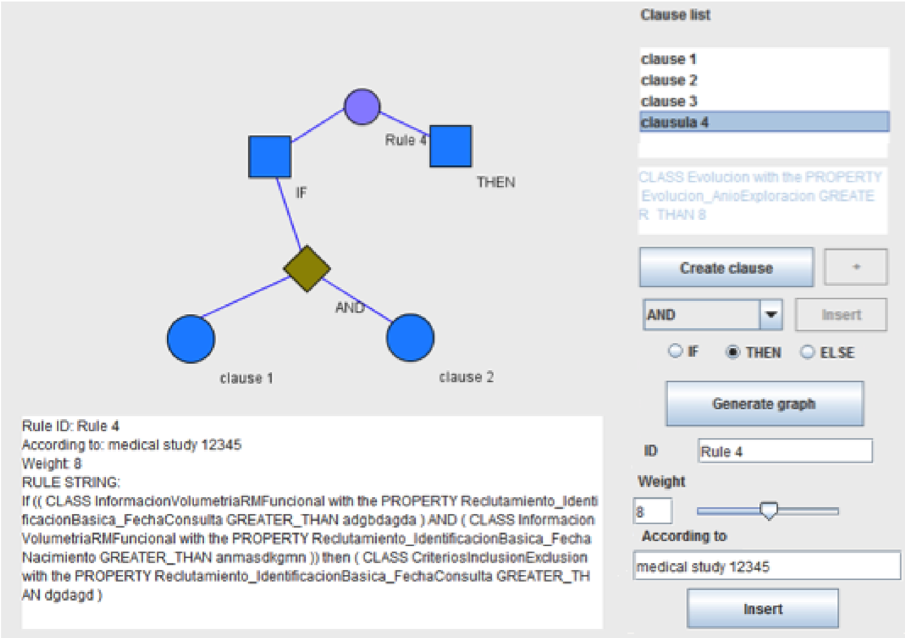

Figure 9.3: Screenshot of the rule generation tool

The rule editor tool stores rules into a xml document that follows a syntax similar to RuleML [BW01]. In our implementation, rules follow an IF-THEN-ELSE structure and are both, (i) weighted within an importance hierarchy of rules, and (ii) endorsed by the corresponding bibliographic source. Each rule in the xml document is stored into a main branch called "LoadRule" (child of the root branch "RuleSet") divided in 5 child-branches:

(i) "RuleID", containing the identifier of the rule,

(ii) "Rule", containing the description of the rule, as follows: "if (conditional clauses) then (consequent clauses) else (alternative consequent clauses)", where a clause is described as:

"CLASS (class_name) with the PROPERTY (datatype_property_name) (modifier) (value)", being the possible modifiers: "EQUALS TO", "SMALLER THAN", "GREATER THAN",

(iii) "weight", containing a ranking number between 0 and 1, and

(iv) "AccordingTo", containing the bibliographic reference of the rule (following SWAN).

The rule corresponding to the example provided above (in step 2) with the conditions for the MAMA-50 radiotherapic protocol is shown in Appendix A.

Implementation of the recommendations generation process

To allow the interaction between the user and the system, a web application has been implemented (Application agent), based on struts2[3], html[4], Javascript[5], jQuery[6] and Jetty[7]. Physicians log in to a web application and access their list of assigned patients. For each patient, corresponding data (variables) can be accesed. An example screenshot of the patient data GUI is shown in Figure 9.4. It is composed by (i) data inherent to the patient or coming from medical results, and also by (ii) data requiring decision processes (marked in Figure 9.4 with a magnifying glass). The latter conform the list of different decision domains for which the system provides recommendations.

[3]Apache Struts2 Home Page: `goo.gl/HkdvQW`
[4]HTML2 Working Group Home Page: `goo.gl/Hq4qqJ`
[5]W3C JavaScript Web APIs: `goo.gl/hMIAiU`
[6]jQuery Home Page: `goo.gl/oWB9ck`
[7]Eclipse Jetty Home Page: `goo.gl/WOoFCb`

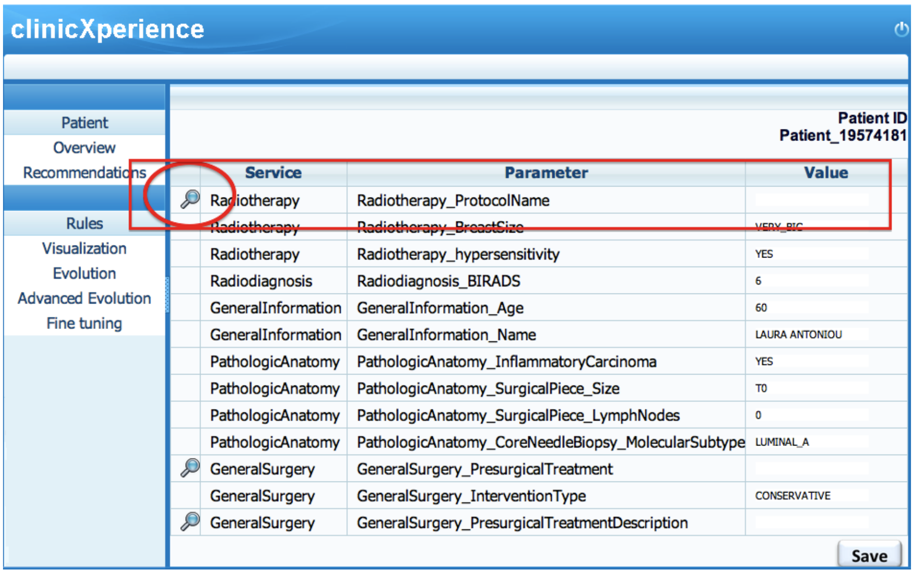

Figure 9.4: Screenshot of an example GUI of patient data

Physicians can optionally request decision recommendations for a certain decision cathegory. Three different features are provided to them: (a) patient relevant data summary, (b) recommendation options with their corresponding percentages (shown in Figure 9.5) and a graphical pie chart plot, and (c) bibliography attached to each recommendation option.

Finally, physicians select a final decision value, included or not in the set of recommended options. At this point the Experience acquisition and Handling agent formalizes the decision event to a SOE serializing variables and rules into the structure, and adding it to the DDNA in the Experience repository. For the implementation of SOEKS/DDNA the SOEKS API[8] has been applied. It is a Java-based library that provides the means to create, manipulate and import/export SOEKS into XML or OWL formats.

[8]The SOEKS API has been developed by the Knowledge Engineering Research Team (KERT) from The University of Newcastle, Australia. Visit `goo.gl/TJLf0j` for more information.

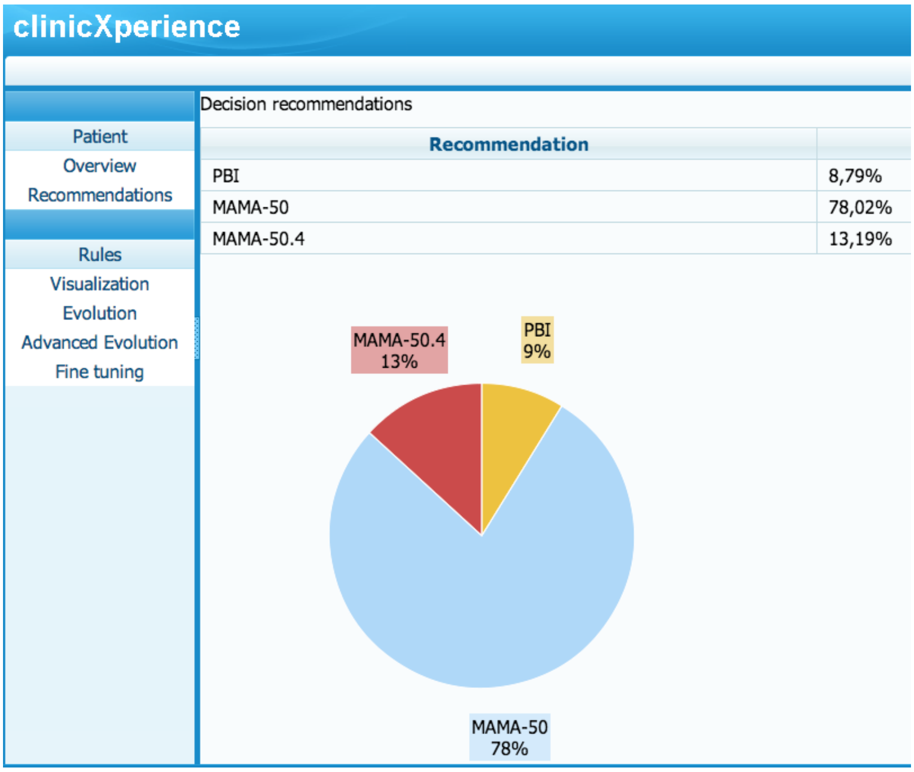

Figure 9.5: Screenshot of an example GUI of recommended options

Implementation of the ruleset evolution

The three different ruleset evolution methods described in CHapter 5 have been implemented: rule weight evolution, new rule generation and fine-tuning of rules. Thus, decisional events stored in the DDNA are reused by the system. Figure 9.6 depicts the resulting weights for three different rules of our implemented ruleset, when the α parameter is set to 0 (i.e. quantitative evolution) (left) and to 1 (i.e. qualitative evolution) (right). It is notorious that rule 17 loses its weight in the second case, as it has error the 100% of times and that rule 2 reduces considerably its weight, due to a high error rate.

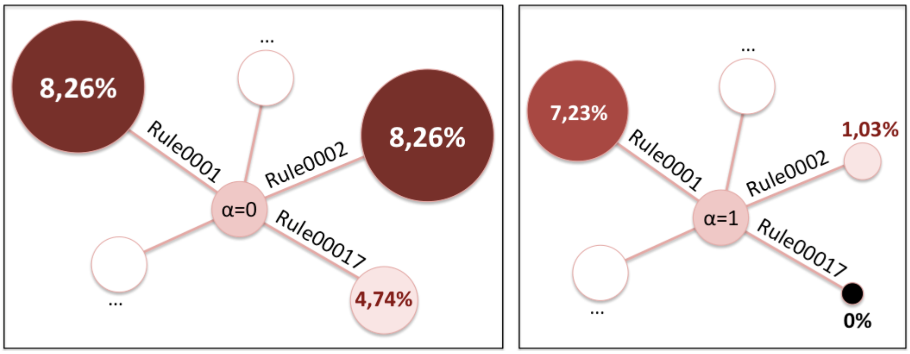

Figure 9.6: Rule weight evolution

Figure 9.7 shows an example screenshot of fine-tuning of rules, where the value T1mic for the surgical piece size is recommended to be removed from the rule. Finally, Figure 9.8 shows an example screenshot of a new rule generation.

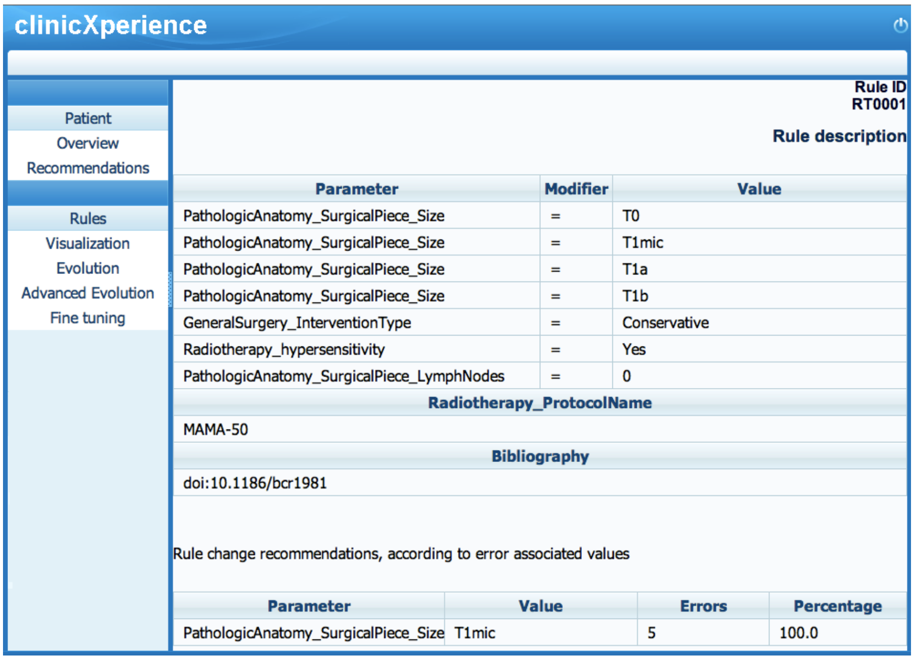

Figure 9.7: Screenshot example of fine-tuning of rules

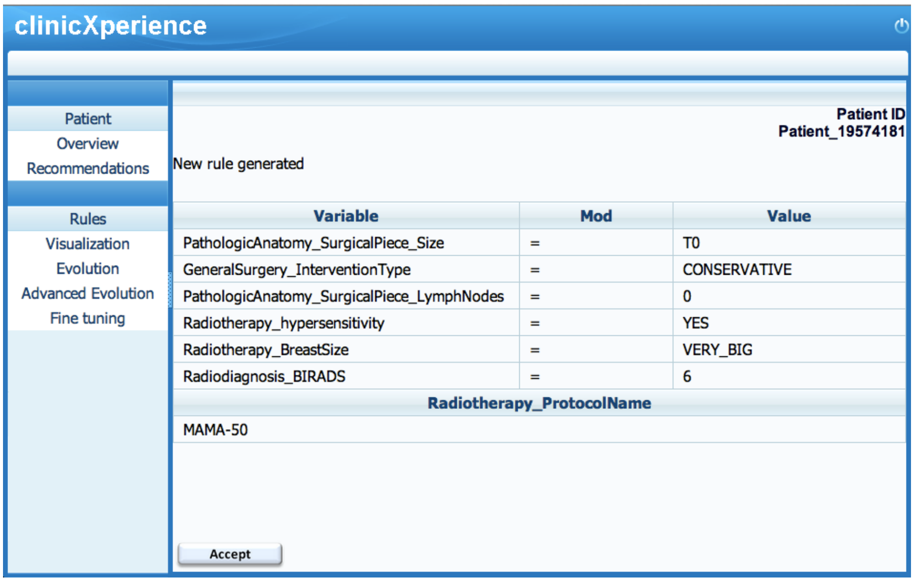

Figure 9.8: Screenshot example of new rule generation

9.6 S-CDSS Evaluation methodology

According to the project schedule, a comprehensive evaluation of the system will be performed during 15 months starting in May 2014, hence the definitive evaluation is outside the scope of this Thesis (altought we plan a future paper reflecting them). The system evaluation methodology [Bür01] that the project will follow consists of four parts: (i) verification, (ii) validation, (iii) evaluation of the human factors, and (iv) evaluation of the clinical effects of the system.

Verification is the process of checking whether the development of the system complies with specifications [Bür01] in terms of provided support for the recommendations. In our case it is trivially done by manual verification.

The **human factor evaluation** process consists of checking the usefulness of the system, its usability, and the satisfaction of the user with the different aspects of the system [Bür01]. Both quantitative and qualitative measures will be obtained. The qualitative analysis will be focused on a questionnaire where physicians can provide their opinion about usability and utility of the system. The results obtained will be studied for improving the system in a future work. The quantitative analysis, on the other side, will be based on a log, storing the number of times physicians have voluntarily accessed the decision support module. These statistics will be compared with the answers in the questionnaires in order to conclude which of the reported factors are in fact the most influential ones.

The **evaluation of the system impact** on the clinical practice will be carried out by statistical analysis of outcome clinical quality indicators (i.e. number of diagnosed patients, number of treated patients, number of recovered patients) for (i) the last 12 months before the LIFE system was integrated in the hospital and (ii) the first 12 months of use of it. Possible external changes in between, such as new medical infrastructure acquired by the hospital or changes in available personnel will be taken into account.

9.6.1 Validation

Validation is the process of checking whether the deveoloped design carries out tasks adequately during a real clinical environment [Bür01]. Let $R_t = \{r_1, r_2, \ldots, r_{Kt}\}$ be the set rules of the KREG Model at time t, which contain the criteria for decision making embedded in the S-CDSS. We will assume that the knowledge embedded in the system is time-varying, and therefore the set of rules may change in time. More specifically, in [TSC+12] we described the possible changes as fine-tuning of rules, deprecation or creation of new rules. Each rule is a tuple $r_k =< a_k, q_k, w_k, b_k >$, where a_k denote the antecedents of the rule, q_k denote the rule consequent, w_k , is a rule weight, and b_k is the bibliographic endorsement. Let $D = \{d_1, d_2, \ldots, d_D\}$ be the set of different decision domains considered in the S-CDSS. Examples of such decision domains are the diagnosis of a patient, the type of treatment prescribed, and the quantity of drug doses, amongst others. For each decision domain d_i the system outputs a collection of decision tuples $o_i = \{dp_{i,1}, dp_{i,2}, \ldots, dp_{i,Ci}\}$. A decision tuple is given by $dp_{i,j} =< c_{i,j}, p_{i,j}, R_{i,j} >$, where $c_{i,j}$, is a selected decision value, $p_{i,j}$, is the probability attached to that decision value, and $R_{i,j}$ is a set of rules that provide the supporting evidence for the aforementioned value. $R_{i,j}$ is formed by the rules r_k whose consequent q_k are equal to the selected decision value $c_{i,j}$, being$R_{i,j} = \{r_k | \, q_k = c_{i,j}\}$.

In the Life system validation two different aspects will be checked: (i) similarity between the output of the system and the final decision made by physicians, and

(ii) reduction of the error as the experience is acquired by the system.

9.6.1.1 Similarity between the output of the system and the final decision of physicians

Both quantitative and qualitative analysis will be carried out in order to validate whether our system infers appropriate results or not. We define for the quantitative analysis a similarity measure $S(f_i, o_i)$ between (i) the system output decision o_i (collection of decision tuples) inferred for each decisional event d_i and (ii) the final decision f_i made by physicians. At this point, the similarity measure compares the decision value selected by the physician versus the output given by the reasoning tool.

As for the experimental design, we will collect data corresponding to o_i and f_i of 1000 decisional events, of 10 different decision domains applied to 100 patients. The similarity $S(f_i, o_i)$ will be calculated for each decisional event. Cases where the normalized similarity value is higher than the 90% will be counted as true positives. As a result of our validation, we will report sensitivity of the system, as well as similarity measure distributions by patient and by decision. The qualitative evaluation will be performed from a short questionnaire filled by the physicians stating their own supporting evidences. This trace of the physician reasoning will be compared with the recommendations of the S-CDSS.

9.6.1.2 Reduction of the error with experience

The experience-based evolution process of rules needs also to be validated. For this purpose after 12 months the same patient data will be re-introduced again in the system. At this time, it is expected that the system will contain an evolved version of the ruleset, R_{t1}, and thus, inferred outputs could differ from the initial ones. Analyzing the new outputs for every decisional event will allow to measure the increasing of similarity with the physician response relative to the initial rule set R_{t0}. From this analysis we will conclude the effectiveness of this agent.

9.7 Discussion - A word on validating

In our approach it is not possible to perform a classical validation, where a training data set is used to build the system and a validation data set is used to evaluate it. This is mainly due to the nature of our system of being based on a knowledge model provided by a team of domain experts. Owing to the aforesaid reasons the correctness cannot be guaranteed with a classical metric. Our system assumes that the model is correct, and we validate this model comparing decisions recommended by the system with decisions made by end-users (physicians), in order to evaluate discrepancies from a real world decision maker solution.

Chapter 10

Conclusions and main contributions

This Chapter summarizes the contributions and conclusions of this dissertation. The Chapter is structured as follows: Section 10.1 discusses the degree of accomplishment of the Thesis objectives and the research question stated in Chapter 1. Section 10.2 summarizes general conclusions. Finally, Section 10.3 discusses some remaining open challenges and future work.

10.1 Research question and objectives accomplishment

Along this dissertation, we have demostrated that clinical experience can be modeled, acquired and reused in the context of clinical decision making. In this Thesis we have proposed a theoretical framework, its specific recommendations, associated methodologies and practical tools allowing the handling of the colective experience within a medical organization throught the semanticaly steered CDSS. In order to ground our contribution, we started by reviewing the most important concepts related to experience and decisional modeling, followed by a review of the current state of art of CDSS. According to the objectives presented in Chapter 1, we have presented a methodology for the generation of decision recommendations. Additionally, we presented a methodology for the acquisition and consolidation of decisional events in the system. In particular, we have provided a methodology for the automatic evolution of a ruleset based on the acquired decicional events. The integration of such contributions into a CDSS allowed us to present an innovative architecture for Semantically steered CDSS (S-CDSS). The architecture fits in the Clinical Task Model (CTM), a generic model for clinical tasks which we also presented. We have achieved an operational implementation of such architecture and methodologies in the framework of two case studies: early diagnosis of AD and breast cancer treatment.

10.2 General conclusions

The main contribution of this work is a framework for semantically steering Clinical Decision Support Systems, in order to (i) support the generation of decision recommendations based on the data and knowledge, (ii) the maintenance of the knowledge in the system, and (iii) the handling of the experience of the clinical team using the system.

Reasoning and recommendation generation

In Chapter 4 the process of recommendation generation carried out by a knowledge based CDSS has been presented. Such process is based on (i) an ontology, containing the semantic models of the domain knowledge and patient data, and (ii) a set of production rules provided by domain experts. Rule antecedents are queried against the ontology, in order to get output recommendations. Such recommendations are then provided to system users as a support for clinical decisions.

In our approach, we have focused on two aspects: (i) speeding-up the reasoning process and (ii) allowing the addition of knowledge generated from the querying process to the ontology. With that purpose, we have applied the Reflexive Ontologies (RO) [Tor08], where queries and the matching individuals are included in the ontology.

As an additional improvement, we have extended RO, adding rules and the corresponding individuals-recommendations to the ontology itself. We have named such approach as Extended Reflexive Ontologies (ROX). ROX, provides a higher speed up of the reasoning process for recommendation generation, as for each rule the set of different recommendations with their corresponding set of individuals is already stored. Thereby, recommendations do not need to be calculated for an individual, but only retrieved directly from the ROX structure. Additionally, the autopoietic behaviour of ROX could provide a framework for analyzing the rules that apply to the mayority of individuals (general rules) or only to a certain minority (specific rules).

In Chapter 8 we have presented an empirical evaluation of the performance of Reflexive Ontologies (RO). In Chapter 7 an implementation of the reasoning and recommendation generation process has been presented, carried out within the framework of the MIND project, for the early diagnosis of Alzheimer's Disease (AD). We have evaluated such implementation, by measuring the speed up provided by RO. In our results we have shown that RO performs much better than a system implemented without RO. However, such results are not generalizable, as they depend on the specific implementation scenario (i.e. the complexity of the ontology, the number of individuals of the ontology, the number of rules, the number of queries per rule, and the query variability between different rules).

Our work is a continuation of the work started by Toro et al. in [Tor08]. At that time, ontology querying tools such as SPARQL [KGH11] were not trustworthy, and the development of a new querying system was required during the implementation of RO. Currently, SPARQL is a consolidated technology with a very efficient implementation. As our work is performed in the framework of different industrial projects, we are in a continous enhancement that will lead towards new implementations and the use of more stable and up-to-date software tools. An example of the aforementioned fact are the efforts (not reflected in this Thesis) towards the serialization of ontologies as N-triples [jgB04] that will certainly help with future scalability and redundancy minimizing when dual data and knowledge repositories are to be created for persistence. N-triples is readable both from Knowledge Bases expressed as ontologies and classic Database approaches.

Experience-based learning and system maintenance

In Chapter 5 we have focused on the S-CDSS knowledge maintenance. Our approach consists of an experience-learning process, where past decisions are acquired and modeled by the system, and then reused to generate new knowledge or modify the existing knowledge. For the modeling and acquisition of the basic experience data structure we have applied the SOEKS/DDNA technologies [CS09, SMASC09]. We have thus proposed a methodology for the evolution of the ruleset, based on previous decisional events. Particularly, we have proposed three different algorithms: (i) an algorithm for rule weight evolution, (ii) an algorithm for fine-tuning of rules, and (iii) a new rule generation algorithm.

In our work, the learning of the system is driven by an error measure, defined as the difference between the recommendation provided by the system and the final choice of the physician. The reduction of rule-weights, the changes on rules, or the generation of new rules is based on the minimization of such error. Therefore, in our approach the evolution converges towards rules that provide recommendations more similar to the final choices of decision makers.

However, such error measure is subjective, as two different physicians could decide for two different decision options for the same patient. In order to apply an objective measure, the effect of a single decision should be traced. A negative evaluation of the effect would then drive an evolution to avoid the same decision in the future. Nevertheless, traceability of decisional events is still not possible, due to the difficulties of measuring of the outcome of a single decisional event.

Additionally, our experience-based learning approach, could also be seen as the validation of the implemented (rules) clinical protocol. In fact, the evolved set of rules will be modeling an updated protocol. Thus, medical teams could apply such system for the validation and maintenance of clinical guidelines and protocols followed during clinical practice.

In Chapter 9 we presented an implementation of our experience-based learning methodologies for a CDSS for Breast Cancer.

A general architecture for Semantically Steered CDSS (S-CDSS)

In Chapter 6 we have presented an architecture for Semantic Steered CDSS (S-CDSS). As the architecture is oriented to covering the whole clinical workflow, we have introduced the Clinical Task Model (CTM). The CTM locates the different clinical information processing stages (i.e. diagnosis, prognosis, treatment, evolution and prevention) in a cyclic chain of federated information processing agents. On the basis of the CTM, we have presented a S-CDSS architecture providing an elegant framework for the integration and reutilization of decision support systems in clinical environments, while answering the main CDSS challenges identified in Chapter 3:

- **Computerized clinical decision support** is provided by a rule system and reasoning engine that infers the corresponding decision recommendations.
- System **extensibility** is guaranteed by the use of a Multi-Agent System architecture providing modularity, scalability and reutilization.
- On the other hand, the acquisition and handling of experience provided by SOEKS/DDNA allows the **maintainability** of the underlying knowledge bases of the S-CDSS.
- Additionally, **timely advice** is provided by the use of Reflexive Ontologies, that speed up reasoning processes and improve the overall efficiency of the system.
- Lastly, the **evaluation of costs and effects** of CDSS is also supported by the handling of the acquired experience.

Clinical workflow integration has also been tackled in Chapter 6, but only at the level of knowledge reutilization amongst the different clinical tasks. Integration of our S-CDSS within clinical systems of hospitals or medical centers is still a challenge. In our approach, we have assumed that variables $\{V_n\}$ can be directly retrieved from data bases and sources. In fact, in both case studies were we have applied S-CDSS (see Chapters 7 and 9), data was provided in a variable-value structure. Hence, loading such data in our system was direct. However, in other clinical environments patient data could be stored in Electronic Health Record (EHR), which follows a different structure and a data storage paradigm. Particularly, patient data is generally stored textually, and thus the extraction of $\{V_n\}$ becomes a natural language processing task. In order to extend the use of our system to a more general case, we consider the development of such natural language processing module as our next step.

Additionally, our proposed architecture for S-CDSS can be set under the category of Service-Model architecture presented by Wright et al. [WS08]. The standard interface in our case is at the side of patient data. If our system is able to extract variables $\{V_n\}$ from such data, the loading and reasoning processes do not need to be done locally. Instead this can be provided via a web service, if security and confidentiality issues are cleared. The most important part in such case would be to guarantee that the knowledge model and the extracted data are aligned. For that purpose, a specific knowledge model for each hospital is needed, which is provided by their experts, but at the same time a standard model could gather the rest of the models together. Such gathering can be performed by a social process, where domain experts can agree on the protocols. Therefore, our S-CDSS architecture could lead to a Framework for Social Clinical Guidelines and Protocols.

10.3 Future Work

In Chapter 8 we have presented an implementation of the reasoning over the query set concept presented in Chapter 4. It allows the extraction of knowledge about the use of different parts of the ontology, enabling a more efficient management of the queries. In order to accomplish that, we have defined an Occurrence Factor F as a quantitative measurement of the recurrence in query related to a certain class in the ontology. As future work we will deepen into the design of more refined algorithms in order to extract implicit knowledge from the reflexive structure, so that the list of RO queries can be optimized by means of pre-caching techniques. Additionally, we plan to extend this work for evaluating the performance of RO in different domains and use cases. In future evaluations we would aim to measure the impact (in terms of computational cost) of Autopoiesis, so that the efficiency of the algorithms could be estimated.

In Chapter 5 we have proposed a methodology for the knowledge maintenance of our system, based on an experience-based learning process that is driven by final decisions made by physicians. As future work, we would like to study how to develop decision traceability in the clinical domain. Such traceability should consider both, the clinical effects of a decision and the economical requirements (in terms of the cost of the treatment, or the use of resources).

Also, knowledge maintenance could be provided by integrating automatic knowledge retrieval tools, able of inporting directly knowledge from existent bibliographic sources and databases. This research line is aligned with Evidence Based Medicine [Str11], where CDSS have been already identified as key tools to support daily practice.

Additionally, a formal evaluation of our architecture in a real clinical environment is still needed, to measure the effects and impact provided to patients healthcare. In order to do so, we are planning a clinical study where the evolution of patients treated with the use of the S-CDSS will be compared with the evolution of similar patients treated in the classical way.

Lastly, the integration of the proposed S-CDSS with hospital EHR is also an open challenge, as EHR are not only clinical data repositories, but also potential medical knowledge repositorioes, from which clinical conclusions and recommendations could be obtained [GPCPMS10]. On the one hand, methodologies for structuring textual clinical data should be developed. On the other hand, EHR semantization efforts should be made. In particular, there are several standards for EHR, such as ISO EN 13606, HL7 CDA and Open EHR, but still a semantic modeling that allows reasoning processes over clinical data has not been achieved yet [LRP11,SC06]. For instance, the works of Smith et al. [SC06] in the semantic modeling of HL7-RIM and Lozano-Rubi et al. [LRP11] in ISO EN 13606, show the limitations and difficulties when approaching such tasks.

Appendix A

Production rule example

The following is an example of a production rule containing conditions for recommending a certain radiotherapic protocol to a Breast Cancer patient.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RuleSet>
<LoadRule>
    <RuleID>RT0001</RuleID>
    <Rule>
    If (( CLASS PathologicAnatomy with the PROPERTY
    PA_SurgicalPiece_Size EQUALS TO T0 ) OR
    ( CLASS PathologicAnatomy with the PROPERTY
    PA_SurgicalPiece_Size EQUALS TO T1mic ) OR
    ( CLASS PathologicAnatomy with the PROPERTY
    PA_SurgicalPiece_Size EQUALS TO T1a ) OR
    ( CLASS PathologicAnatomy with the PROPERTY
    PA_SurgicalPiece_Size EQUALS TO T1b ) AND
    ( CLASS GeneralSurgery with the PROPERTY
    GS_InterventionType EQUALS TO Conservative ) AND
    ( CLASS Radiotherapy with the PROPERTY
    RT_hypersensitivity EQUALS TO Yes ) AND
    ( CLASS PathologicAnatomy with the PROPERTY
```

```
PA_SurgicalPiece_LymphNodes EQUALS TO 0 ))
then ( CLASS Radiotherapy with the PROPERTY
RT_ProtocolName EQUALS TO MAMA-50 )
</Rule>
<weight>1</weight>
<AccordingTo>
    <classes>
        <class>GivenBySpecialist</class>
    </classes>
    <specialist>

        <specialistType>Radiotherapist</specialistType>
        <specialistPlace>
        Hospital General Universitario de Valencia
        </specialistPlace>
        <reportTitle>NE</reportTitle>
    </specialist>
    <contributionAuthors>

        <contributionAuthor>NA</contributionAuthor>
    </contributionAuthors>
    <doi>NA</doi>
    <title>NA</title>
    <contributionEditors>

        <contributionEditor>NA</contributionEditor>
    </contributionEditors>
    <contributionPublisher>NA</contributionPublisher>
    <contributionPublicationDate>NA</contributionPublicationDate>
    <contributionPublicationPlace>NA</contributionPublicationPlace>
```

```
            <isbn10>NA</isbn10>
            <isbn13>NA</isbn13>
            <issn>NA</issn>
            <journalName>NA</journalName>
            <volume>NA</volume>
            <issue>NA</issue>
            <newspaperName>NA</newspaperName>
            <pagination>NA</pagination>
            <websiteName>NA</websiteName>
        </AccordingTo>
    </LoadRule>
    </RuleSet>
```

Bibliography

[AAA$^+$09] Roman Alberca, Guillermo Amer, Carmen Antúnez, Miquel Baquero, Mercè Boada, Hermenegildo J. Bueno, Jesús Cacho, Teresa Calatayud, Jaume Campdelacreu, Cristóbal Carnero, Ignacio Casado, Dobato José L., Sara Fernández, Ana Frank, Rocío García-Ramos, García Guillermo, Justo García, Jordi Gascón, Jose L. González, Pedro Guardado, Isabel Hernández, Pedro E. Jiménez, Adolfo Jiménez, José P. Lara, Albert Lladó, Marcos Llanero, Alberto Lleó, Secundino López, José López, M. Sagrario Manzano, Pablo Martínez, M. Dolores Martínez, Manuel Menéndez, Jaume Morera, Rubén Muñiz, Javier Olazarán, M. Ángel Ponce, Ramón Reñé, Norberto Rodríguez, José Rojo, Jose M. Ruiz, Raquel Sánchez, Carmen Terrón, Eduardo Torrealba, Meritxell Valentí, and Félix Viñuela. *Guía oficial para la práctica clínica de demencias: conceptos, criterios y recomendaciones.* Number 8 in Guías Oficiales de la Sociedad Española de Neurología. Prois Science (Thomson Reuters), Barcelona, Spain, 2009.

[AAHS07] Samina Raza Abidi, Syed Sibte Raza Abidi, Sajjad Hussain, and Mike Shepherd. Ontology-based modeling of clinical practice guidelines: A clinical decision support system for breast cancer follow-up interventions at primary care settings. In Klaus A. Kuhn, James R. Warren, and Tze-Yun Leong, editors, *MedInfo*, volume 129 of *Studies in Health Technology and Informatics*, pages 845–849. IOS Press, 2007.

[ABF12] Mobyen Uddin Ahmed, Shahina Begum, and Peter Funk. Case studies on the clinical applications using case-based reasoning. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *FedCSIS*, pages 3–10, 2012.

[AG97] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9(7):1545–1588, 1997.

[AH03] Grigoris Antoniou and Frank Van Harmelen. Web ontology language: Owl. In *Handbook on Ontologies in Information Systems*, pages 67–92. Springer, 2003.

[ALK+02] Mehmet Altinel, Qiong Luo, Sailesh Krishnamurthy, C. Mohan, Hamid Pirahesh, Bruce G. Lindsay 0001, Honguk Woo, and Larry Brown. Dbcache: database caching for web application servers. In Michael J. Franklin, Bongki Moon, and Anastassia Ailamaki, editors, *SIGMOD Conference*, page 612. ACM, 2002.

[Ami05] McIlraith S. Amir, E. Partition-based logical reasoning for first-order and propositional theories. *Artificial Intelligence*, 162:49–88, 2005.

[AP94] Agnar Aamodt and Enric Plaza. Case-based reasoning; foundational issues, methodological variations, and system approaches. *AI COMMUNICATIONS*, 7(1):39–59, 1994.

[APT03] Khalil Amiri, Sanghyun Park, and Renu Tewari. Dbproxy: A dynamic data cache for web applications. In *In Proc. ICDE*, pages 821–831, 2003.

[Art11] Arkaitz Artetxe. Meta api genérica para la gestión y representación computacional de grafos. Master's thesis, University of the Basque Country, April 2011.

[ASL11] L. Aleksovska-Stojkovska and S. Loskovska. Architectural and data model of clinical decision support system for managing asthma in school-aged children. In *Electro/Information Technology (EIT), 2011 IEEE International Conference on*, pages 1–6, 2011.

[AWP05] José F. Arocha, Dongwen Wang, and Vimla L. Patel. Identifying reasoning strategies in medical decision making: A methodological guide. *Journal of Biomedical Informatics*, 38(2):154–171, 2005.

[BCHH87] G O Barnett, J J Cimino, J A Hupp, and E P Hoffer. Dxplain. an evolving diagnostic decision-support system. *JAMA*, 258(1):67–74, Jul 1987.

[BD03] Michael Boahene and George Ditsa. Knowledge management. chapter Conceptual Confusions in Knowledge Management and Knowledge Management Systems: Clarifications for Better KMS Development, pages 12–24. IGI Global, Hershey, PA, USA, 2003.

[Ber68] Ludwig von Bertalanffy. *General Systems Theory. Foundations, Development, Applications.* George Braziller, New York, 1968.

[Ber04] R. Bergmann. *Experience Management.* Springer, New York, 2004.

[BHJV08] Jürgen Bock, Peter Haase, Qiu Ji, and Raphael Volz. Benchmarking OWL Reasoners. In Frank van Harmelen, Andreas Herzig, Pascal Hitzler, Zuoquan Lin, Ruzica Piskac, , and Guilin Qi, editors, *Proceedings of the ARea2008 Workshop*, volume 350, http://ceur-ws.org, June 2008. CEUR Workshop Proceedings.

[BJB+05] L. Backman, S. Jones, A. K. Berger, E. J. Laukka, and B. J. Small. Cognitive impairment in preclinical Alzheimer's disease: a meta-analysis. *Neuropsychology*, 19:520–531, Jul 2005.

[BKW+03] David W Bates, Gilad J Kuperman, Samuel Wang, Tejal Gandhi, Anne Kittler, Lynn Volk, Cynthia Spurr, Ramin Khorasani, Milenko Tanasijevic, and Blackford Middleton. Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inform Assoc*, 10(6):523–530, Nov-Dec 2003.

[BL04] Stefano Borgo and Paulo Leitão. The role of foundational ontologies in manufacturing domain applications. In Robert Meersman and Zahir Tari, editors, *CoopIS/DOA/ODBASE (1)*, volume 3290 of *Lecture Notes in Computer Science*, pages 670–688. Springer, 2004.

[BL07] Eta S. Berner and Tonya J. La Lande. *Clinical Decision Support Systems, Theory and Practice*, chapter Overview of Clinical Decision Support Systems. Number 1. Springer, New York, second edition, 2007.

[Ble69] Howard L. Bleich. Computer evaluation of acid-base disorders. *The Journal of Clinical Investigation*, 48(9):1689–1696, 9 1969.

[BLL+91] T. A. Brennan, L. L. Leape, N. M. Laird, L. Hebert, A. R. Localio, A. G. Lawthers, J. P. Newhouse, P. C. Weiler, and H. H. Hiatt. Incidence of adverse events and negligence in hospitalized patients. results of the harvard medical practice study i. *The New England journal of medicine*, 324(6):370–376, February 1991.

[Blo12] B Blomqvist. The use of semantic web technologies for decision support - a survey. *Semantic Web journal*, Undefined:1–24, 2012.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Bre01] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[BRH09] Matt-Mouley Bouamrane, Alan L. Rector, and Martin Hurrell. Development of an ontology for a preoperative risk assessment clinical decision support system. In *CBMS*, pages 1–6. IEEE, 2009.

[BS03] L. S. Bickley and P.G. Szilagyi. *Bates' Guide to Physical Examination and History Taking*. Lippincott Williams and Wilkins, Philadelphia, 8 edition, 2003.

[BTL+99] D W Bates, J M Teich, J Lee, D Seger, G J Kuperman, N Ma'Luf, D Boyle, and L Leape. The impact of computerized physician order entry on medication error prevention. *J Am Med Inform Assoc*, 6(4):313–321, Jul-Aug 1999.

[Bür01] Ammenwerth E. Prokosch H.U. Dudeck J. Bürkle, T. Evaluation of clinical information systems. what can be evaluates and what cannot? *Journal of Evaluation in Clinical Practice*, 7(4):373–385, 2001.

[BW01] S. Tabet H. Boley and G. Wagner. Design rationale of RuleML: A markup language for semantic web rules. In I. F. Cruz, S. Decker, J. Euzenat, and D. L. McGuinness, editors, *Proc. Semantic Web Working Symposium*, pages 381–402, Stanford University, California, 2001.

[BW11] A. Banerjee and T.F. Watson. *Pickard's Manual of Operative Dentistry*, chapter Chapter 3: Diagnosis, prognosis, care planning: 'information processing'. Oxford University Press, 2011.

[BW12] Archie Bleyer and H. Gilbert Welch. Effect of three decades of screening mammography on breast-cancer incidence. *New England Journal of Medicine*, 367(21):1998–2005, 2012. PMID: 23171096.

[CCG91] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *Neural Networks, IEEE Transactions on*, 2(2):302–309, 1991.

[CDV08] Michele Ceccarelli, Antonio Donatiello, and Dante Vitale. Kon3: A clinical decision support system, in oncology environment, based on knowledge management. *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, 2:206–210, 2008.

[CGC13] B. Blobel C. Gonzalez C, D.M. Lopez. Case-based reasoning in intelligent health decision support systems. *Studies in Health Technology and Informatics*, 189:44–59, 2013.

[CHW06] Michael J. Collins, Jeffrey Hoffmeister, and Steven W. Worrell. Computer-aided detection and diagnosis of breast cancer. *Seminars in Ultrasound, {CT} and {MRI}*, 27(4):351 – 355, 2006. <ce:title>Breast Imaging Update</ce:title>.

[Cio01] J Cioffi. A study of the use of past experiences in clinical decision making in emergency situations. *International Journal of Nursing Studies*, 38(5):591 – 599, 2001.

[CLHL12] Jiuwen Cao, Zhiping Lin, Guang-Bin Huang, and Nan Liu. Voting based extreme learning machine. *Information Sciences*, 185(1):66 – 77, 2012.

[CM10a] S. Herrera Delgado D. Rodriguez García C. Maurente, E. Ocampo Edye. *Innovations and Advances in Computer Sciences and Engineering*, chapter Evaluation of Case Based Reasoning for Clinical Decision Support Systems applied to Acute, pages 259–264. Springer, 2010.

[CM10b] Leslie Crews and Eliezer Masliah. Molecular mechanisms of neurodegeneration in Alzheimer's disease. *Human Molecular Genetics*, 19(1):12–20, Apr 2010.

[CN11] P. Croskerry and GR Nimmo. Better clinical decision making and reducing diagnostic error. *The Journal of the Royal College of Physicians of Edinburght*, 42:155–162, 2011.

[Cor03] Daniel D Corkill. Collaborating software: Blackboard and multi-agent systems and the future. In *Proceedings of the International Lisp Conference*, New York, New York, October 2003.

[Cra95] I. Craig. *Blackboard Systems*. Ablex Publishing, Norwood, US, 1995.

[CS07] E. Szczerbicki C. Sanin. Towards the construction of decisional dna: A set of experience knowledge structure java class within an ontology system. *Cybernetics and Systems: An International Journal*, 38:859–878, 2007.

[CS09] E. Szczerbicki C. Sanin. Experience-based knowledge representation soeks. *Cybernetics and Systems: An International Journal*, 40(2):99–122, 2009.

[CSJ+10] H.D. Cheng, Juan Shan, Wen Ju, Yanhui Guo, and Ling Zhang. Automated breast cancer detection and classification using ultrasound images: A survey. *Pattern Recognition*, 43(1):299 – 317, 2010.

[CTS+08] Yolanda Cobos, Carlos Toro, Cristina Sarasua, Javier Vaquero, María Teresa Linaza, and Jorge Posada. An architecture for fast semantic retrieval in the film heritage domain. In Ebroul Izquierdo, editor, *CBMI*, pages 272–279. IEEE, 2008.

[CWW+08] Paolo Ciccarese, Elizabeth Wu, Gwen Wong, Marco Ocana, June Kinoshita, Alan Ruttenberg, and Tim Clark. The {SWAN} biomedical discourse ontology. *Journal of Biomedical Informatics*, 41(5):739 – 751, 2008. <ce:title>Semantic Mashup of Biomedical Data</ce:title>.

[dABW+98] Klaus dieter Althoff, Ralph Bergmann, Stefan Wess, Michel Manago, Eric Auriol, Oleg I. Larichev, Er Bolotov, Yurii I. Zhuravlev, and Serge I. Gurov. Case-based reasoning for medical decision support tasks: The inreca approach. In *Artificial Intelligence in Medicine 12*, pages 25–41, 1998.

[DBF+13] C. Dromain, B. Boyer, R. Ferre, S. Canale, S. Delaloge, and C. Balleyguier. Computed-aided diagnosis (cad) in the detection of breast cancer. *European Journal of Radiology*, 82(3):417 – 423, 2013. <ce:title>Breast Imaging</ce:title>.

[DC06] S. De Castro. *Manual de Patología General.* ELSEVIER-MASSON, Barcelona, Spain., 6 edition, 2006.

[dDLS+72] F. T. de Dombal, D. J. Leaper, J. R. Staniland, A. P. McCann, and Jane C. Horrocks. Computer-aided diagnosis of acute abdominal pain. *BMJ*, 2(5804):9–13, 4 1972.

[DE10] M. Das and J. Eichner. Challenges and barriers to clinical decision support (cds) design and implementation experienced in the agency for healthcare research and quality cds demonstrations. Technical Report 10-0064-EF, AHRQ Publication, Rockville, 2010.

[dPCVOV+09] Jesus de Pedro-Cuesta, Javier Virues-Ortega, Saturio Vega, Manuel Seijo-Martinez, Pedro Saz, Fernanda Rodriguez, Angel Rodriguez-Laso, Ramon Rene, Susana Perez de las Heras, Raimundo Mateos, Pablo Martinez-Martin, Jose Maria Manubens, Ignacio Mahillo-Fernandez, Secundino Lopez-Pousa, Antonio Lobo, Jordi Llinas Regla, Jordi Gascon, Francisco Jose Garcia, Manuel Fernandez-Martinez, Raquel Boix, Felix Bermejo-Pareja, Alberto Bergareche, Julian Benito-Leon, Ana de Arce, and Jose Luis del Barrio. Prevalence of dementia and major dementia subtypes in spanish populations: a reanalysis of dementia prevalence surveys, 1990-2008. *BMC Neurol*, 9:55, 2009.

[DR05] A. Hassane D. Rene. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, Berlin, 2005.

[Dru95] P. Drucker. *The Post-Capitalist Executive: Managing in a Time of Great Change*. Penguin, New York, 1995.

[dVRS+08] E N de Vries, M A Ramrattan, S M Smorenburg, D J Gouma, and M A Boermeester. The incidence and nature of in-hospital adverse events: a systematic review. *Quality and Safety in Health Care*, 17(3):216–223, 2008.

[EA04] H.M. Ghaziri E.M. Awad. *Knowledge Management*. Prentice Hall, New Jersey, 2004.

[Els09] ArthurS. Elstein. Thinking about diagnostic thinking: a 30-year perspective. *Advances in Health Sciences Education*, 14(1):7–18, 2009.

[EPC+98] R S Evans, S L Pestotnik, D C Classen, T P Clemmer, L K Weaver, J F Jr Orme, J F Lloyd, and J P Burke. A computer-assisted management program for antibiotics and other antiinfective agents. *N Engl J Med*, 338(4):232–238, Jan 1998.

[ES02] Arthur S Elstein and Alan Schwarz. Clinical problem solving and diagnostic decision making: selective review of the cognitive literature. *BMJ*, 324(7339):729–732, 3 2002.

[FGJ97] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proc. Symposium on Ontological Engineering of AAAI*, 1997.

[FM83] Edward Feigenbaum and Pamela McCorduck. *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1983.

[FM99] R.A. Flores-Mendez. Towards the standardization of multi-agent system frameworks: an overview. *ACM Crossroads, Special Issue on Intelligent Agents, Association for Computer Machinery*, 5(4):18–24, 1999.

[For10] Pyne J.M. Steven C.A. Williams J.S. Hedrick R.G. Lunsford A.K. Raney W.N. Ackerman B.A. Ducker L.O. Bonner L.M. Smith J.L Fortney, J.C. A web-based clinical decision support system for depression care management. *Am J Manag Care*, 16(11):849–854, 2010.

[FS07] G. Fung and J. Stoeckel. SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. *Knowl. Inf. Syst.*, 11(2):243–258, 2007.

[FW05] H. H. Feldman and M. Woodward. The staging and assessment of moderate to severe Alzheimer disease. *Neurology*, 65(6 Suppl 3):S10–S17+, 2005.

[Gér08] Starfield B. Heath I. Gérvas, J. Is clinical prevention better than cure? *The Lancet*, 372:1997–1999, 2008.

[GG95] N. Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32, 1995.

[GGL+90] R M Gardner, O K Golubjatnikov, R M Laub, J T Jacobson, and R S Evans. Computer-critiqued blood ordering using the help system. *Comput Biomed Res*, 23(6):514–528, Dec 1990.

[GM98] A Geissbuhler and R A Miller. Clinical application of the umls in a computerized order entry and decision-support system. *Proc AMIA Symp*, pages 320–324, 1998.

[GPCPMS10] Carlos Gallego-Perez, Joan Cornet-Prat, and Josep Manyach-Serra. Standards for interoperability: new challenges. *Medicina Clinica*, 134(1):32–38, 2010.

[GPFC04] A. Gómez-Pérez, M. Fernández, and O. Corcho. *Ontological Engineering*. Springer-Verlag, London, 2004.

[GPH04] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. An evaluation of knowledge base systems for large owl datasets. In *In International Semantic Web Conference*, pages 274–288. Springer, 2004.

[GPSK05] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Automatic partitioning of owl ontologies using e-connections. In *In International Workshop on Description Logics*, 2005.

[Gra93] John B. E. Atwood M. E. Gray, W. D. Project ernestine: Validating a goms analysis for predicting and explaining real-world performance. *Human-Computer Interaction*, 8(3):237–309, 1993.

[Gre06] R.A Greenes. Why clinical decision support is hard to do. *AMIA AnnuSymp Proc. 2006*, page 1169, 2006.

[Gre11] R.A. Greenes. *Clinical Decision Support: The Road Ahead*. Elsevier Science, 2011.

[Gro13] U.S. Cancer Statistics Working Group. United states cancer statistics: 1999–2010 incidence and mortality web-based report. Technical report, Department of Health and Human Services, Centers for Disease Control and Prevention and National Cancer Institute, Atlanta: U.S, 2013.

[Gru95] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5–6):907–928, 1995.

[Has11] Isaac RK. Hasan, SS. An integrated approach of mas-commonkads, model-view-controller and web application optimization strategies for web-based expert system development. *Expert Systems with Applications: An International Journal*, 38(1):417–428, 2011.

[Hay98] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edition, July 1998.

[HD03] R.L Hayward and J.M Dixon. Current limits of knowledge in adjuvant and neoadjuvant endocrine therapy of breast cancer: the need for more clinical research. *Surgical Oncology*, 12(4):289 – 304, 2003. Breast Cancer.

[HDB95] M.T. Hagan, H.B. Demuth, and M.H. Beale. *Neural Network Design*. PWS Pub. Co., Har/Dsk edition, December 1995.

[HH06] H. Herre and B. Heller. Semantic foundations of medical information systems based on top-level ontologies. *Knowledge-Based Systems*, 19(2):107 – 115, 2006. <ce:title>Intelligent Software Design</ce:title>.

[HK11] Jairo Robledo Jr. H. Kashfi. Towards a case-based reasoning method for openehr-based clinical decision support. In *Proceedings of The 3rd International Workshop on Knowledge Representation for Health Care (KR4HC'11)*, Bled, Slovenia, July 2011.

[Ho98] T.K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998.

[Hol08] C.W. Holsapple. *Decision Support System Fundamentals*, volume II of *Handbook on Decision Support Systems*, chapter DSS Architecture and Types, pages 163–189. Springer Berlin Heidelberg, 2008.

[Hor11] Matthew Horridge. *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. The University Of Manchester, 1.3 edition, March 2011.

[Hül07] E. Hüllermeier. *Case-Based Approximate Reasoning*. Springer-Verlag, Berlin, 2007.

[HXB03] A. Holbrook, S. Xu, and J. Banting. What factors determine the success of clinical decision support systems? *AMIA AnnuSympProc 2003*, page 862, 2003.

[HZS06] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489 – 501, 2006.

[IA] D. Isern and Moreno A. Computer-based execution of clinical guidelines: A review. *International Journal of Medical Informatics*, 77:787–808.

[Ise10] Sanchez D. Moreno A. Isern, D. Agents applied in health care: A review. *International Journal of Medical Informatics*, 79:145–166, 2010.

[jgB04] jan grant and Dave Beckett. Rdf test cases. W3c recommendation, W3C, 2004. http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/.

[JMM+13] Afsaneh Jalalian, Syamsiah B.T. Mashohor, Hajjah Rozi Mahmud, M. Iqbal B. Saripan, Abdul Rahman B. Ramli, and Babak Karasfi. Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review. *Clinical Imaging*, 37(3):420 – 426, 2013.

[Joh07] M. Johnstone. Patient safety ethics and human error management in ed contexts part i: development of the global patient safety movement. *Australasian Emergency Nursing Journal*, 10(1):13–20, 2007.

[JTMP01] P D Johnson, S W Tu, M A Musen, and I Purves. A virtual medical record for guideline-based decision support. *Proc AMIA Symp*, pages 294–298, 2001.

[Kal03] Carson E.R. Collison P.O. Kalogeropoulos, D.A. Towards knowledge-based systems in clinical practice: Development of an integrated clinical information and knowledge management support system. *Computer Methods and Programs in Biomedicine*, 72:65–80, 2003.

[Kas10] Jerome P. Kassirer. Teaching clinical reasoning: case-based and coached. *Academic medicine : journal of the Association of American Medical Colleges*, 85(7):1118–1124, jul 2010.

[Kaw07] Lobach D.F. Kawamoto, K. Proposal for fulfilling strategic objectives of the u.s. roadmap for national action on decision support through a service-oriented architecture leveraging hl7 services. *Journal of the American Medical Informatics Association*, 14(2):146–155, 2007.

[KC72] Dankward Kodlin and Morris F. Collen. Automated diagnosis in multiphasic screening, 1972.

[KCD00] L.T. Kohn, J. Corrigan, and M.S. Donaldson. *To err is human: building a safer health system*. National Academy Press, Washington, 2000.

[KF11] Kuan-Liang Kuo and Chiou-Shann Fuh. A rule-based clinical decision model to support interpretation of multiple data in health examinations. *J. Medical Systems*, 35(6):1359–1373, 2011.

[KGH11] Ilianna Kollia, Birte Glimm, and Ian Horrocks. Sparql query answering over owl ontologies. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'11, pages 382–396, Berlin, Heidelberg, 2011. Springer-Verlag.

[KGP12] Gilad J. Kuperman, Reed M. Gardner, and T. Allan Pryor. *HELP: A Dynamic Hospital Information System*. Springer Publishing Company, Incorporated, 1st edition, 2012.

[KHBL05] K. Kawamoto, C.A. Houlihan, E.A. Balas, and D.F. Lobach. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ*, 330(7494):765, 2005.

[KL05] Kensaku Kawamoto and David F Lobach. Design, implementation, use, and preliminary evaluation of sebastian, a standards-based web service for clinical decision support. *AMIA Annu Symp Proc*, pages 380–384, 2005.

[Koh89] T. Kohonen. *Self-organization and associative memory: 3rd edition*. Springer-Verlag New York, Inc., 1989.

[KRW+95] C E Jr Kahn, L M Roberts, K Wang, D Jenks, and P Haddawy. Preliminary investigation of a bayesian network for mammographic diagnosis of breast cancer. *Proc Annu Symp Comput Appl Med Care*, pages 208–212, 1995.

[KXY08] Guilan Kong, Dong-Ling Xu, and Jian-Bo Yang. Clinical decision support systems: A review on knowledge representation and inference under uncertainties. *International Journal of Computational Intelligence Systems*, 1(2):159–167, 2008.

[LKA+07] David F Lobach, Kensaku Kawamoto, Kevin J Anstrom, Michael L Russell, Peter Woods, and Dwight Smith. Development, deployment and usability of a point-of-care decision support system for chronic disease management using the recently-approved hl7 decision support service standard. *Stud Health Technol Inform*, 129(Pt 2):861–865, 2007.

[LL59] R S LEDLEY and L B LUSTED. Reasoning foundations of medical diagnosis; symbolic logic, probability, and value theory aid our understanding of how physicians reason. *Science*, 130(3366):9–21, Jul 1959.

[LMC+06] Hugo Y. K. Lam, Luis Marenco, Tim Clark, Yong Gao, June Kinoshita, Gordon Shepherd, Perry Miller, Elizabeth Wu, Gwen Wong, Nian Liu, Chiquito Crasto, Thomas Morse, Susie Stephens, and Kei hoi Cheung. Semantic web meets e-neuroscience: An rdf use case. In *In Proceedings of the ASWC International Workshop on Semantic e-Science*, pages 158–170. University Press, 2006.

[LRP11] Raimundo Lozano-Rubi and Xavier Pastor. Building a clinical repository based on iso-13606. In *Proceeding of the third Symposium on Healthcare Systems Interoperability*, 2011.

[LT+09] O. Luaces, F. Taboada, et al. Predicting the probability of survival in intensive care unit patients from a small number of variables and training examples. *Artificial Intelligence in Medicine*, 45(1):63–76, January 2009. PMID: 19185475.

[LWA06] Joseph Liu, Jeremy C Wyatt, and Douglas G Altman. Decision tools in health care: focus on the problem, not the solution. *BMC Med Inform Decis Mak*, 6(4), Jan 2006.

[MAB06] B. H. Monien, L. G. Apostolova, and G. Bitan. Early diagnostics and therapeutics for Alzheimer's disease–how early can we get there? *Expert Rev Neurother*, 6(9):1293–1306, Sep 2006.

[Mar10] W. Marwan. Amoeba-inspired network design. *Science*, 327(5964):419–20, 2010.

[Mat80] Varela F.J. Maturana, H.R. *Autopoiesis and Cognition: The Realization of the Living*, volume 42 of *Boston Studies in the Philosophy and History of Science*. Springer, 1980.

[Mat09] Ledeczi A. Nadas A. Sztipanovits J. Martin J.B. Weavind L.M. Miller A. Miller P. Maron D.J. Mathe, J.L. A model-integrated, guideline-driven. *Clinical Decision-Support System. Software IEEE*, 26(4):54 – 61, 2009.

[MFS+96] P L Miller, S J Frawley, F G Sayward, W A Yasnoff, L Duncan, and D W Fleming. Imm/serve: a rule-based program for childhood immunization. *Proc AMIA Annu Fall Symp*, pages 184–188, 1996.

[MG12] I. Marques and M. Graña. Face recognition with lattice independent component analysis and extreme learning machines. *Soft Computing*, on line, February 2012.

[Mil83] P L Miller. Critiquing anesthetic management: the "attending" computer system. *Anesthesiology*, 58(4):362–369, Apr 1983.

[MMJ+77] C J McDonald, R Murray, D Jeris, B Bhargava, J Seeger, and L Blevins. A computer-based record and clinical monitoring system for ambulatory care. *Am J Public Health*, 67(3):240–245, Mar 1977.

[MPM82] R A Miller, H E Jr Pople, and J D Myers. Internist-1, an experimental computer-based diagnostic consultant for general internal medicine. *N Engl J Med*, 307(8):468–476, Aug 1982.

[MSE08] Joy Higgs Megan Smith and Elizabeth Ellis. *Clinical reasoning in the health professions*, chapter Factors influencing clinical decision making, pages 89–100. Butterworth Heinemann, Amsterdam, 3 edition, 2008.

[MYG+13] Ashutosh Malhotra, Erfan Younesi, Michaela Gündel, Bernd Müller, Michael T. Heneka, and Martin Hofmann-Apitius. Ado: A disease ontology representing the domain knowledge specific to alzheimer's disease. *Alzheimer's and Dementia*, (0):–, 2013.

[MYQ+06] Li Ma, Yang Yang, Zhaoming Qiu, Guotong Xie, Yue Pan, and Shengping Liu. Towards a complete OWL ontology benchmark. In *Proceedings of the 3rd European conference on The Semantic Web: research and applications*, ESWC'06, pages 125–139, Berlin, Heidelberg, 2006. Springer-Verlag.

[NGM14] Argha Nandy, Sudeshna Gangopadhyay, and Ashis Mukhopadhyay. Individualizing breast cancer treatment; the dawn of personalized medicine. *Experimental Cell Research*, 320(1):1 – 11, 2014.

[NJR14] Mehdi Namdari and Hooshang Jazayeri-Rad. Incipient fault diagnosis using support vector machines based on monitoring continuous decision functions. *Engineering Applications of Artificial Intelligence*, 28(0):22 – 35, 2014.

[NM01] Natalya F. Noy and Deborah L. Mcguinness. Ontology development 101: A guide to creating your first ontology. Technical report, 2001.

[NSW+09] Natalya F. Noy, Nigam H. Shah, Patricia L. Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L. Rubin, Margaret-Anne Storey, Christopher G. Chute, and Mark A. Musen. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 2009.

[NVN+10] Mikael Nystrom, Anna Vikstrom, Gunnar H. Nilsson, Hans Ahlfeldt, and Håkan Orman. Enriching a primary health care version of icd-10 using snomed ct mapping. *Journal of biomedical semantics*, 1(1):7, 2010.

[OAH+07] Daniel Oberle, Anupriya Ankolekar, Pascal Hitzler, Philipp Cimiano, Michael Sintek, Malte Kiesel, Babak Mougouie, Stephan Baumann, Shankar Vembu, Massimo Romanelli, Paul Buitelaar, Ralf Engel, Daniel Sonntag, Norbert Reithinger, Berenike Loos, Hans-Peter Zorn, Vanessa Micelli, Robert Porzel, Christian Schmidt, Moritz Weiten, Felix Burkhardt, and Jianshen Zhou. {DOLCE} ergo sumo: On foundational and domain models in the smartweb integrated ontology (swinto). *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(3):156 – 174, 2007.

[OF10] Ramaswamy M. Ortiz-Fournier, L. Healthcare knowledge collection for clinical decision support systems. *Issues in Information Systems*, 11(1):538–546, 2010.

[OMGM+98] L Ohno-Machado, J H Gennari, S N Murphy, N L Jain, S W Tu, D E Oliver, E Pattison-Gordon, R A Greenes, E H Shortliffe, and G O Barnett. The guideline interchange format: a model for representing guidelines. *J Am Med Inform Assoc*, 5(4):357–372, Jul-Aug 1998.

[oR13] American College of Radiology. *ACR BIRADS Atlas*. ACR, 5th edition edition, 2013.

[OTM+07] J.A. Osheroff, J.M. Teich, B.F. Middleton, E.B. Steen, A. Wright, and D.E. Detmer. A roadmap for national action on clinical decision support. *JAMIA*, 14(2):141–145, 2007.

[Ple10] M. Pleckaityte. Alzheimer's disease: a molecular mechanism, new hypotheses, and therapeutic strategies. *Medicina (Kaunas)*, 46:70–76, 2010.

[Pow08] D.J Power. *Decision Support Systems: A Historical Overview*, chapter Part I – Foundation of Decision Support Systems, pages 121–140. Handbook on Decision Support Systems. Springer, 2008.

[Pre08] Jean E. Pretz. Intuition versus analysis: strategy and experience in complex everyday problem solving. *Memory & cognition*, 36(3):554–566, apr 2008.

[PT06] M. Peleg and S.W. Tu. *2006 IMIA Yearbook of Medical Informatics: Assessing Information - Technologies for Health*, chapter Decision support, knowledge representation and management in medicine, pages 72–80. Schattauer Verlagsgesellschaft GmbH, Stuttgart, 2006.

[PW12] E. Szczerbicki P. Wang, C. Sanin. Introducing the concept of decisional dna based web content mining. *Cybernetics and Systems: An International Journal*, 37:97 – 117, 2012.

[RBT+04] Prabhu Ram, David Berg, Samson Tu, Guy Mansfield, Qin Ye, Robert Abarbanel, and Nick Beard. Executing clinical practice guidelines using the sage execution engine. *Stud Health Technol Inform*, 107(Pt 1):251–255, 2004.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, pages 318–362. MIT Press, 1986.

[Rin07] A. Rinaldi. Science and technology are looking to nature's successful designs for inspiration. *EMBO reports*, 8(11):995, 2007.

[Rin11] Carmeli B. Kent C. Landau D. Maman Y. Rubin Y. Slonim N. Rinott, R. Prognostic data-driven clinical decision support - formulation and implications. In Andersen S.K. Aarts J. Hurlen P. Moen, A., editor, *Studies in Health Technology and Informatics 169, User Centred Networked Health Care - Proceedings of MIE*, pages 140–144, 2011.

[Roz06] C. Rozman. *Compendio de Medicina Interna*. Elsevier, Madrid, Spain, 3 edition, 2006.

[RR00] Roe C. Rasmequan, S. and S.B. Russ. Strategic decision support systems: An experience-based approach. In *18th IASTED Conference on Applied Informatics*, pages 14–17, Innsbruck, Austria, February 2000.

[RSL+12] Lorenzo M. Refolo, Heather Snyder, Charlene Liggins, Laurie Ryan, Nina Silverberg, Suzana Petanceska, and Maria C. Carrillo. Common alzheimer's disease research ontology: National institute on aging and alzheimer's association collaborative project. *Alzheimer's and Dementia*, 8(4):372 – 375, 2012.

[SB] Alan Schwartz and George Bergus. *Medical Decision Making: A Physician's Guide*. Cambridge University Press, New York.

[SBH+06] P. Schloeffel, T. Beale, G. Hayworth, S. Heard, and H. Leslie. The relationship between CEN 13606, HL7, and openEHR. *HIC 2006 and HINZ 2006: Proceedings*, 24, 2006.

[SC06] Barry Smith and Werner Ceusters. Hl7 rim: An incoherent standard. In Arie Hasman, Reinhold Haux, Johan van der Lei, Etienne De Clercq, and Francis H. Roger France, editors, *MIE*, volume 124 of *Studies in Health Technology and Informatics*, pages 133–138. IOS Press, 2006.

[SC09] C. Tappert S.H. Cha. A genetic algorithm for constructing compact binary decision trees. *Journal of Pattern Recognition Research*, 4(1):1–13, 2009.

[SCGR+11] J. G. Stone, G. Casadesus, K. Gustaw-Rothenberg, S. L. Siedlak, X. Wang, X. Zhu, G. Perry, R. J. Castellani, and M. A. Smith. Frontiers in Alzheimer's Disease Therapeutics. *Ther Adv Chronic Dis*, 2:9–23, Jan 2011.

[SDA+75] E H Shortliffe, R Davis, S G Axline, B G Buchanan, C C Green, and S N Cohen. Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the mycin system. *Comput Biomed Res*, 8(4):303–320, Aug 1975.

[SF04] Z. Sun and G.R. Finnie. *Intelligent Techniques in E-Commerce: A Case Based Reasoning Perspective.* Studies in Fuzziness and Soft Computing. Springer, 2004.

[SG03] Zhaohao Sun and Finnie G. Brain-like architecture and experience-based reasoning. In *Proc. 7th Joint Conf on Information Sciences (JCIS)*, pages 1735–1738, Cary, North Carolina, USA, September 2003.

[SGP+89] D F Sittig, R M Gardner, N L Pace, A H Morris, and E Beck. Computerized management of patient care in a complex, controlled clinical trial in the intensive care unit. *Comput Methods Programs Biomed*, 30(2-3):77–84, Oct-Nov 1989.

[SK99] P. Somervuo and T. Kohonen. Self-Organizing maps and learning vector quantization for feature sequences. *Neural Process. Lett.*, 10(2):151–159, 1999.

[SKDO06] Katharina Schwarz, Timo Kouwenhoven, Virginia Dignum, and Jacco Van Ossenbruggen. Supporting the decision process for the choice of a domain modeling scheme. In *In Workshop 'Formal Ontologies Meet Industry', Castelnuovo del Garda*, pages 9–10, 2006.

[SMASC09] Cesar Sanín, Leonardo Mancilla-Amaya, Edward Szczerbicki, and Paul CayfordHowell. Application of a multi-domain knowledge structure: The decisional dna. In Raymond Chiong and Sandeep Dhakal, editors, *Intelligent Systems for Knowledge Management*, volume 250 of *Studies in Computational Intelligence*, pages 65–86. Springer, 2009.

[SP09] David Sutton and Vivek Patkar. Commonkads analysis and description of a knowledge based system for the assessment of breast cancer. *Expert Systems with Applications*, 36(2, Part 1):2411 – 2423, 2009.

[Spe90] Donald F. Specht. Probabilistic neural networks. *Neural Netw.*, 3(1):109–118, 1990.

[SSB+81] Edward H. Shortliffe, A. Carlisle Scott, Miriam B. Bischoff, A. Bruce Campbell, William van Melle, and Charlotte D. Jacobs. Oncocin: An expert system for oncology protocol management. In Patrick J. Hayes, editor, *IJCAI*, pages 876–881. William Kaufmann, 1981.

[SSF+11] William W. Stead, John R. Searle, Henry E. Fessler, Jack W. Smith, and Edward H. Shortliffe. Biomedical informatics: changing what physicians need to know and how they learn. *Academic medicine : journal of the Association of American Medical Colleges*, 86(4):429–434, apr 2011.

[SSG12] Neeraj Sharma, Kawaljeet Singh, and D. P. Goyal. Is technology universal panacea for knowledge and experience management? answers from indian it sector. In Sumeet Dua, Aryya Gangopadhyay, Parimala Thulasiraman, Umberto Straccia, Michael A. Shepherd, and Benno Stein, editors, *ICISTM*, volume 285 of *Communications in Computer and Information Science*, pages 187–198. Springer, 2012.

[SST07] Cesar Sanín, Edward Szczerbicki, and Carlos Toro. An owl ontology of set of experience knowledge. *Journal of Universal Computer Science*, pages 209–223, February 2007.

[STA+12] Eider Sanchez, Carlos Toro, Arkaitz Artetxe, Manuel Graña, Eduardo Carrasco, and Frank Guijarro. A semantic clinical decision support system: conceptual architecture and implementation guidelines. In *KES*, pages 1390–1399, 2012.

[STC+11a] Eider Sanchez, Carlos Toro, Eduardo Carrasco, Patricia Bonachela, Carlos Parra, Gloria Bueno, and Frank Guijarro. A knowledge-based clinical decision support system for the diagnosis of alzheimer disease. In *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services (Healthcom 2011)*, pages 355–361, Columbia, Missouri, USA, 2011. IEEE, IEEE.

[STC+11b] Eider Sanchez, Carlos Toro, Eduardo Carrasco, Gloria Bueno, Carlos Parra, Patricia Bonachela, Manuel Graña, and Frank Guijarro. An architecture for the semantic enhancement of clinical decision support systems. In *KES (2)*, pages 611–620, 2011.

[Str11] Glasziou P. Richardson W.S. Haynes R.B. Straus, S.E. *Evidence-based medicine. How to practice and teach it.* Elsevier, 4 edition, 2011.

[STZ+12] Cesar Sanín, Carlos Toro, Haoxi Zhang, Eider Sanchez, Edward Szczerbicki, Eduardo Carrasco, Peng Wang, and Leonardo Mancilla-Amaya. Decisional dna: A multi-technology shareable knowledge structure for decisional experience. *Neurocomputing*, 88:42–53, 2012.

[SWO+08] D.F. Sittig, A. Wright, J.A. Osheroff, B. Middleton, J.M. Teich, J.S. Ash, E. Campbell, and D.W Bates. Grand challenges in clinical decision support. *Journal of biomedical informatics*, 41(2):387–392, 2008.

[Syc98] K.P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.

[TA14] Shokoufe Tayyebi and Maryam Alishiri. The control of {MSF} desalination plants based on inverse model control by neural network. *Desalination*, 333(1):92 – 100, 2014.

[TDE+12] Jill Elizabeth Thistlethwaite, David Davies, Samilia Ekeocha, Jane M. Kidd, Colin MacDougall, Paul Matthews, Judith Purkis, and Diane Clay. The effectiveness of case-based learning in health professional education. a beme systematic review: Beme guide no. 23. *Medical Teacher*, 34(6):e421–e444, 2012. PMID: 22578051.

[Ter10] A. Tero. Rules for biologically inspired adaptive network design. *Science*, 327(5964):439, 2010.

[TGP+09] Carlos Toro, Manuel Graña, Jorge Posada, Javier Vaquero, and Edward Szczerbicki. Domain modeling based on engineering standards. pages 95–102, September 2009.

[Tor08] Sanín C. Szczerbicki E. Posada J. Toro, C. Reflexive ontologies: Enhancing ontologies with self-contained queries. *Cybernetics and Systems: An International Journal*, 39:171–189, 2008.

[TSC+12] Carlos Toro, Eider Sanchez, Eduardo Carrasco, Leonardo Mancilla-Amaya, Cesar Sanín, Edward Szczerbicki, Manuel Graña, Patricia Bonachela, Carlos Parra, Gloria Bueno, and Frank Guijarro. Using set of experience knowledge structure to extend a rule set of clinical decision support system for alzheimer's disease diagnosis. *Cybernetics and Systems*, 43(2):81–95, 2012.

[TSF+93] J M Teich, C D Spurr, S J Flammini, J Schmiz, R F Beckley, J F Hurley, M Aranow, and J P Glaser. Response to a trial of physician-based inpatient order entry. *Proc Annu Symp Comput Appl Med Care*, pages 316–320, 1993.

[TSV+07] Carlos Toro, Cesar Sanín, Javier Vaquero, Jorge Posada, and Edward Szczerbicki. Knowledge based industrial maintenance using portable devices and augmented reality. pages 295–302, September 2007.

[TTLW06] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for Support Vector Machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1088–1099, 2006.

[TVS+07] Carlos Toro, Javier Vaquero, Cesar Sanín, Edward Szczerbicki, and Jorge Posada. Implementing decisional dna in industrial maintenance by a knowledge soupa extension. *Systems Science, Wroclaw University of Technology, Poland*, pages 61–68, September 2007.

[Vap98] V. N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, September 1998.

[War89] H R Jr Warner. Iliad: moving medical decision-making into new frontiers. *Methods Inf Med*, 28(4):370–372, Nov 1989.

[Whe11] Noy N.F. Shah N.H. Alexander P.R. Nyulas C. Tudorache T. Musen M.A. Whetzel, P.L. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Res*, 39(Web Server issue):541–5, 2011.

[WPT+04] Dongwen Wang, Mor Peleg, Samson W Tu, Aziz A Boxwala, Omolola Ogunyemi, Qing Zeng, Robert A Greenes, Vimla L Patel, and Edward H Shortliffe. Design and implementation of the glif3 guideline execution engine. *J Biomed Inform*, 37(5):305–318, Oct 2004.

[WS08] A. Wright and D.F. Sittig. A four-phase model of the evolution of clinical decision support architectures. *International Journal of Medical Informatics*, 77(10):641 – 649, 2008.

[WW04] Anna Wills and Ian Watson. Building a case-based reasoner for clinical decision support. In Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors, *PRICAI*, volume 3157 of *Lecture Notes in Computer Science*, pages 554–562. Springer, 2004.

[WWK13] Andreas Wicht, Thomas Wetter, and Ulrike Klein. A web-based system for clinical decision support and knowledge maintenance for deterioration monitoring of hemato-oncological patients. *Computer Methods and Programs in Biomedicine*, 111(1):26–32, 2013.

[Xin10] Shengjun Z. Xing, L. One mountain with two tigers – china and the united states in east asian regionalism. *Perspectives of federalism*, 2(3):110–129, 2010.

[Zai13] D.A. Zaitsev. Toward the minimal universal petri net. *IEEE Transactions on Systems Man and Cybernetics*, 99:1–12, 2013.

[ZHCZ13] Xutang Zhang, Xin Hou, Xiaofeng Chen, and Ting Zhuang. Ontology-based semantic retrieval for engineering domain knowledge. *Neurocomputing*, 116(0):382 – 391, 2013. <ce:title>Advanced Theory and Methodology in Intelligent Computing</ce:title> <ce:subtitle>Selected Papers from the Seventh International Conference on Intelligent Computing (ICIC 2011).</ce:subtitle>.

[Zho08] Wang W. Zhou, Q. The automatic inference of arden medical logic modules. In *Proceedings of the international conference on BioMedical Engineering and Informatics*, 2008.