COMBINING ONTOLOGIES AND RULES WITH CLINICAL ARCHETYPES

PhD THESIS SUBMITTED TO THE UNIVERSITY OF ALCALÁ

PROGRAM: INTERDISCIPLINARY APPLICATIONS AND PROBLEMS OF THE INFORMATION AND COMMUNICATION TECHNOLOGIES

PROGRAMA: APLICACIONES Y PROBLEMAS INTERDISCIPLINARES DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

2011

By

Leonardo Lezcano Matías

Supervisor

Dr. Miguel-Ángel Sicilia

University of Alcalá

Contents

Abstract 11

Resumen 13

Dedication 17

Acknowledgments 19

1 Introduction 23

1.1 The Research Agenda 24

1.1.1 Research Objectives 26

1.1.2 Research Contributions 27

1.2 Notations & Vocabulary 28

1.3 Thesis Outline 30

2 Background 33

2.1 Interoperability in health systems 35

2.2 The two-level paradigm and the openEHR approach 39

2.2.1 The openEHR Reference Model, Archetypes and Templates 42

2.2.2 The Archetype Definition Language and the AOM 49

2.3 Semantic Web Ontologies and Rules 54

2.3.1 The Protégé-OWL API 56

2.3.2 SWRL Rules 59

2.4 Clinical Terminologies 62

2.4.1 Relevant Clinical Terminologies and Ontologies 64

2.4.2 Integrating archetypes and clinical terminologies 67

3 Related Work 69

3.1 The POSEACLE approach 70

3.2 The ARTEMIS project 71

3.3 Clinical guidelines integration 72
3.4 Binding terminologies to clinical archetypes 75

4 Overall approach to the translation 77

4.1 Overview of the ADL to OWL translation 77
4.2 User-defined datatypes to represent restrictions on quantified values 82
4.3 Representing *occurrences* restrictions in OWL 84
4.4 A reuse technique based on annotation properties 87
4.5 Constraining possible values to a finite and small set 88
4.6 Translation Reference 91
4.6.1 Data ADL (dADL) to OWL 92
4.6.2 Constraint ADL (cADL) to OWL 93

5 The ADL2OWL translator implementation 95

5.1 The graphic user interface package 100
5.2 The parser package 101
5.3 The translator package 103
5.3.1 The translator.content_translator package 105
5.3.2 The translator.structure_translator package 107
5.3.3 The translator.data_translator package 109
5.3.4 The translator.representation_translator 112

6 Integrating rules with clinical archetypes 115

6.1 Sharing and Reusing SWRL rules 116
6.2 Rules integration and inference execution 118
6.2.1 Translating involved archetypes 120
6.2.2 Defining *linker rules* and *guideline rules* 123
6.2.3 Mapping patients' data to OWL instances 125
6.2.4 SWRL edition and execution 127

7 Bindings to Clinical Ontologies 131

7.1 Ontology navigability 133
7.2 Checking the mapping context 134
7.3 Knowledge Integration 135

8 Evaluation 139

8.1 Decision support in the IEF-EHRS research project....140
8.1.1 The antibiotic prescribing and allergy detection case study....142
8.2 Alerts in the CISEP project....147
8.2.1 The pilot service for pressure ulcer prevention....148
8.3 Other current projects and directions....153
8.3.1 The Clinical DSS research and development project....153
8.3.2 Diagnosing the Obstructive Chronic Pulmonary Disease....156

9 Conclusions 159

10 Conclusiones 165

11 Future Work 171

11.1 Sharing archetypes as linked data....171
11.2 Implementing clinical guidelines and managing archetypes....172
11.3 The ArchOnt framework scope....173

12 References 175

List of Figures

Figure 1. Conventional versus Clinical Data warehouses. Taken from Pedersen and Jensen (1998). 34

Figure 2. Separation of models of information and models of reality. Adapted from Beale and Heard (2008a). 41

Figure 3. The openEHR approach to interoperability and the two-level model shaded in gray. Taken from Garde et al. (2007). 43

Figure 4. The openEHR Health Computing Platform. Adapted from Beale and Heard (2008a). 44

Figure 5. The boundary between the RM and the AM. Adapted from Beale and Heard (2008a). 45

Figure 6. The RM `ENTRY` package as defined by Beale et al. (2008). 47

Figure 7. The openEHR Clinical Knowledge Manager (*archetypes repository*) currently contains near 300 archetype definitions. 48

Figure 8. ADL header section. 50

Figure 9. ADL definition section (*data constraints*). 51

Figure 10. ADL definition section (*state constraints*). 51

Figure 11. ADL definition section (*protocol constraints*). 51

Figure 12. ADL ontology section. 52

Figure 13. The `Constraint_model` package as defined by Beale (2007). 53

Figure 14. Integration of Jena in Protégé-OWL 57

Figure 15. Class diagram of the Protégé-OWL model. 58

Figure 16. *Heart rate and rhythm* archetype (ADL root). 80

Figure 17. *Heart rate and rhythm* archetype (OWL XML/RDF syntax). 80

Figure 18. *Heart rate and rhythm* archetype (OWL Manchester syntax). 80

Figure 19. Restriction of inherited properties to guarantee OWL and ADL structure compatibility. 81

Figure 20. A DV_COUNT constraint from the *Pulse* archetype (ADL). 83

Figure 21. A DV_COUNT constraint from the *Pulse* archetype (OWL XML/RDF syntax). 84

Figure 22. A DV_COUNT constraint from the *Pulse* archetype (OWL Manchester syntax). 84

Figure 23. occurrences restrictions in the *Heart rate and rhythm* archetype (ADL). 85

Figure 24. A constraint on the items cardinality (OWL XML/RDF syntax). 85

Figure 25. ADL occurrences statements expressed in OWL Manchester Syntax. 86

Figure 26. use_node keyword in the *Palpation* archetype. 87

Figure 27. Reusing the *Body_site* class in the *Tenderness* CLUSTER (OWL XML/RDF syntax). 88

Figure 28. Reusing the *Body_site* class in the *Tenderness* CLUSTER (OWL Manchester syntax). 88

Figure 29. The Category ELEMENT in *Palpation* archetype. 89

Figure 30. Specialisation of CODE_PHRASE to represent *Tenderness* categories (OWL XML/RDF syntax). 90

Figure 31. Specialisation of CODE_PHRASE to represent *Tenderness* categories (OWL Manchester syntax). 90

Figure 32. XML Schema built-in datatypes. 91

Figure 33. The ADL to OWL graphic user interface. 96

Figure 34. Structure chart of the ADL2OWL translator. 97

Figure 35. Dependencies between main packages. 98

Figure 36. Class diagram of the translator package. 99

Figure 37. Example Domain-specific AOM classes. 109

Figure 38. The ArchOnt Framework. 118

Figure 39. ADL to OWL translation of the *Intravascular Pressure* archetype. 123

Figure 40. Monitoring *Sinus Ligation* (SWRL context preparation I). 124

Figure 41. Monitoring *Sinus Ligation* (SWRL context preparation II) 124

Figure 42. SWRL representation of the assessment of *Sinus Ligation*. 125

Figure 43. Bindings examples in the *Blood Pressure* archetype. 132

Figure 44. Protégé and OWLViz view of archetypes term bindings represented by equivalence relations. 133

Figure 45. Archetypes connected through the SNOMED-CT ontology. 134

Figure 46. Archetypes and SNOMED-CT contribution in defining the *Drug Prescription* and the *Adverse Reaction* concepts. 135

Figure 47. The screenshot shows the contribution of *Physiology* ontology and *Blood Pressure* archetype in defining the *Systolic Blood Pressure* concept. The properties in the bottom are provided by the *Physiology* ontology while the ones in the top come from the archetype. 136

Figure 48. Tentative mapping between OGMS and openEHR archetypes. 137

Figure 49. OGMS-openEHR integration approach. 138

Figure 50. The IEF-EHRS project conceptual architecture. 141

Figure 51. Antibiotic prescribing - SWRL rule (1st fragment). 143

Figure 52. Antibiotic prescribing - SWRL rule (2nd fragment). 144

Figure 53. Antibiotic prescribing - SWRL rule (3rd fragment). 144

Figure 54. Allergy detection - SWRL rule. 145

Figure 55. Control flow for the chained execution of SWRL rules as an aid to decision making in antibiotic prescribing and allergy detection. 147

Figure 56. The `OBSERVATION`.*Norton* archetype as defined by the *Fuenlabrada Hospital*. 152

Figure 57. Linker rule to fill the *Total Points* `ELEMENT`. 152

Figure 58. SWRL clinical rules for pressure ulcer risk assessment. 153

Figure 59. SWRL rules to infer the stage of COPD according to the GOLD guidelines. 156

Abstract

Like many other fields that heavily rely on the capabilities of information and communication technologies, healthcare and biomedical environments are rapidly increasing the demand for widely accepted agreements on data, information and knowledge exchange. Such needs for compatibility or interoperability go beyond syntactical and structural issues as semantic interoperability is also required. Semantic interoperability is essential to facilitate the computerized support for alerts, workflow management and evidence-based healthcare across heterogeneous Electronic Health Record (EHR) systems.

The model of clinical archetypes supported by the CEN/ISO EN13606 standard and the openEHR foundation provides a mechanism to express data structures in a shared and interoperable way. It has acquired considerable acceptance in the last years by allowing the definition of clinical concepts based on a common Reference Model while low level storage implementation can keep its heterogeneity across EHR systems. However, archetype languages do not provide direct support neither for clinical rules nor mappings to formal ontologies, which are both key elements of full semantic interoperability as they allow exploiting reasoning on clinical knowledge.

It has been acknowledged that the World Wide Web demands analogous capabilities to those mentioned above, leading to the development of the Semantic Web extension. The progress made in that field, regarding reasoning and knowledge

representation, is combined in this thesis with EHR models in order to enhance the archetype approach and to support features that correspond to a richer level of semantic interoperability.

Concretely, this research presents and evaluates an approach to translate definitions expressed in openEHR Archetype Definition Language (ADL) to a formal representation using ontology languages. The approach is implemented in the *ArchOnt* framework, which is also described. The integration of those formal representations with clinical rules is then studied, providing an approach to reuse reasoning on concrete instances of clinical data. Sharing the knowledge expressed in the form of rules is coherent with the philosophy of *open sharing* underlying archetypes, and it also extends reuse to propositions of declarative knowledge as those encoded for example in clinical guidelines. Thus, this thesis describes the techniques to map archetypes to formal ontologies and how rules can be attached to the resulting representation. In addition, the translation allows specifying logical bindings to equivalent clinical concepts from other knowledge sources. Such bindings encourage reuse as well as ontology reasoning and navigability across different ontologies.

Another significant contribution of the thesis is the application of the presented approach as part of two research projects in collaboration with teaching hospitals in Madrid. Examples taken from those cases, such as the development of alerting systems aimed at improving patient safety, are here explained. Besides the direct applications described, the automatic translation of archetypes to an ontology language fosters a wide range of semantic and reasoning activities to be designed and implemented on top of a common representation instead of taking an ad-hoc approach.

Resumen

Al igual que otros campos que dependen en gran medida de las funcionalidades ofrecidas por las tecnologías de la información y las comunicaciones (IT), la biomedicina y la salud necesitan cada vez más la implantación de normas y mecanismos ampliamente aceptados para el intercambio de datos, información y conocimiento. Dicha necesidad de compatibilidad e interoperabilidad va más allá de las cuestiones sintácticas y estructurales, pues la interoperabilidad semántica es también requerida. La interoperabilidad a nivel semántico es esencial para el soporte computarizado de alertas, flujos de trabajo y de la medicina basada en evidencia cuando contamos con la presencia de sistemas heterogéneos de Historia Clínica Electrónica (EHR).

El modelo de arquetipos clínicos respaldado por el estándar CEN/ISO EN13606 y la fundación openEHR ofrece un mecanismo para expresar las estructuras de datos clínicos de manera compartida e interoperable. El modelo ha ido ganando aceptación en los últimos años por su capacidad para definir conceptos clínicos basados en un Modelo de Referencia común. Dicha separación a dos capas permite conservar la heterogeneidad de las implementaciones de almacenamiento a bajo nivel, presentes en los diferentes sistemas de EHR. Sin embargo, los lenguajes de arquetipos no soportan la representación de reglas clínicas ni el mapeo a ontologías formales, ambos elementos fundamentales para alcanzar la interoperabilidad semántica completa pues permiten llevar a cabo el razonamiento y la inferencia a partir del conocimiento clínico existente.

Paralelamente, es reconocido el hecho de que la World Wide Web presenta requisitos análogos a los descritos anteriormente, lo cual ha fomentado el desarrollo de la Web Semántica. El progreso alcanzado en este terreno, con respecto a la representación del conocimiento y al razonamiento sobre el mismo, es combinado en esta tesis con los modelos de EHR con el objetivo de mejorar el enfoque de los arquetipos clínicos y ofrecer funcionalidades que se corresponden con nivel más alto de interoperabilidad semántica.

Concretamente, la investigación que se describe a continuación presenta y evalúa un enfoque para traducir automáticamente las definiciones expresadas en el lenguaje de definición de arquetipos de openEHR (ADL) a una representación formal basada en lenguajes de ontologías. El método se implementa en la plataforma *ArchOnt*, que también es descrita. A continuación se estudia la integración de dichas representaciones formales con reglas clínicas, ofreciéndose un enfoque para reutilizar el razonamiento con instancias concretas de datos clínicos. Es importante ver como el acto de compartir el conocimiento clínico expresado a través de reglas es coherente con la filosofía de intercambio abierto fomentada por los arquetipos, a la vez que se extiende la reutilización a proposiciones de conocimiento declarativo como las utilizadas en las guías de práctica clínica. De esta manera, la tesis describe una técnica de mapeo de arquetipos a ontologías, para luego asociar reglas clínicas a la representación resultante. La traducción automática también permite la conexión formal de los elementos especificados en los arquetipos con conceptos clínicos equivalentes provenientes de otras fuentes como son las terminologías clínicas. Dichos enlaces fomentan la reutilización del conocimiento clínico ya representado, así como el razonamiento y la navegación a través de distintas ontologías clínicas.

Otra contribución significativa de la tesis es la aplicación del enfoque mencionado en dos proyectos de investigación y desarrollo clínico, llevados a cabo en combinación con hospitales universitarios de Madrid. En la explicación se incluyen ejemplos de las aplicaciones más representativas del enfoque como es el caso del desarrollo de sistemas de alertas orientados a mejorar la seguridad del paciente. No obstante, la traducción automática de arquetipos clínicos a lenguajes de ontologías constituye una base común para la implementación de una amplia gama de actividades semánticas, razonamiento y validación, evitándose así la necesidad de aplicar distintos enfoques ad-hoc directamente sobre los arquetipos para poder satisfacer las condiciones de cada contexto.

Dedication

To my grandmother Marina, who is no longer with us.

Acknowledgments

Writing an acknowledgment is quite difficult for most people, for me too. There are several people who have supported me by providing encouragement and useful advice the years of research. Please accept my deepest gratitude, irrespective of your name appearing on present acknowledgement.

I would like to start by thanking my supervisor, Dr. Miguel Angel Sicilia, for leading me through the doctoral programme in a way that has sharpened my skills as a researcher. This is one period in life where a good guidance is essential to reach substantial results. I cannot thank him enough for his constant support and help during this time. I am also grateful to Sicilia for lifting my spirits when things seemed to go wrong, and for those endless discussions which often escaped the scope of this thesis, touching life topics. He also deserves my gratitude for his collaboration in paper publishing and for providing me the opportunities to attend several conferences, workshops and meetings that have widen my research network.

I also thank all the members of the Information Engineering Research Unit for their help and for creating such a good working environment. The jokes and smiles have given a relaxing touch to overcome stressing moments. In addition, I acknowledge both technical support and facilities that they have offered me.

A special thanks to the Open University of Madrid (UDIMA) for allowing me to be part of the important academic project that has been carried out in the last two years. I deeply appreciate the help and advices I have received from all the staff and professors at UDIMA, as they allowed me to combine these years of academic work with the PhD degree. Besides, I must acknowledge that the challenging courses I have had the opportunity to impart at UDIMA have considerably enriched my teaching experiences.

Heartfelt thanks to my wife, Rosmary Calzadilla, who lived every moment of my PhD process with me. For being there through all the ups and downs, for being patient and understand me. Her commitment and sacrifice make me proud to be part of her life. A word of thanks to my entire family, who has contributed so much to my education. All the memories and anecdotes that I keep from them constitute a great source of strength and motivation. I am very grateful to my aunt Laura for introducing me to the wonderful world of computer science when I was barely a child. I would also like to thank all my friends with whom I have shared so many good times. A special thanks to my friend Michel for keeping in touch all these years in spite of the distance. He has always been a reference on how to overcome difficulties and keep moving forward. I would also like to thank my math professor, Carlos Jiménez, whose teachings went beyond mathematics reaching ethical values. I will never forget him.

This acknowledgment cannot be complete without thanking my parents, for being so close to me, for always trusting me and for giving it all for me. They have encouraged my development as student and as a sociable person. They have taught me the principles about what is right and what is wrong, as well as to never give up. I am grateful to both of them, eternally, for making me who I am.

The intensity of the conviction that a hypothesis
is true has no bearing on whether it is true or not.

P.B. Medawar
Advice to a Young Scientist

1 Introduction

Clinical practice can be represented as an iterative care delivery process that starts with *observations* of the status of the patient. Such *observations* lead to informed *opinions* on the part of a health care professional, including assessment of the current situation, goals for a future situation and plans for achieving the goals. Such plans then turn into detailed *instructions* for clinical practice that eventually trigger the appropriate *actions*. At this stage, we may need to repeat the whole iteration until the problem is solved. These four kinds of information are breakpoints where communication between independent healthcare systems is frequently lost because of data ambiguity and incompatibility.

In the last years, the paradigm of archetypes (Garde, Knaup, Hovenga, & Heard, 2007) has brought a new way to define the models of electronic health records and to normalize the information transfer between such heterogeneous healthcare systems, making them more interoperable. Archetypes are formal clinical specifications, expressed in terms of constraints on a generic reference model. They can be combined together through templates, and used at runtime to extract data, to enable querying, and to support legacy data transformation. Thus, archetypes serve as a shared language for common and specialised clinical concepts. In other words, the reference model encloses the stable features like the set of classes that make up the blocks constituting an electronic health record and the basic syntax of statements, while archetypes allow for sharing a wide variety of combinations of those classes corresponding to record fragments created for specific clinical situations. For example, *Blood Pressure*, *Medication Order* and *Transfusion* are clinical statements that have already been specified as archetypes, so they can be used or refined as reference data structures for the interchange of clinical data.

However, improving interoperability requires different systems to have the ability to exchange every possible information related to healthcare, including propositions of *declarative knowledge* as those encoded for example in clinical guidelines. The archetypes paradigm and the underlying two-level model (which are detailed in the Background Chapter) allow reaching this understanding at syntactic, structural and semantic levels (Garde et al., 2007). Nevertheless, when it comes to clinical rules, archetypes and the languages used to define them show themselves insufficient, at least to guarantee a seamless exchange of the underlying semantics.

1.1 The Research Agenda

The consideration of semantic interoperability introduces the need for computational semantics. For example, a health care information system that receives some observation entry like *Body Temperature* (no matter from where but conforming to some archetype specification) should be able to deliver it to the appropriate professional, who would eventually proceed to deal with the assessment of the observation. This is clearly a significant advance for the interoperability of health systems, but it could be further enhanced with semantics attached to the archetypes. If the archetype is linked to knowledge representations, then the system would be able to act upon the information directly (e.g. by triggering an alert or notification), or to suggest the clinician some courses or action or relationships with other existing information. This additional processing on the data requires shared representations as those that can be found in formal *ontologies* (Gruber, 1993). A further step would be that of being able to infer knowledge on how to assess, evaluate and act upon the information, using rules or other automated reasoning systems.

As clinical archetypes currently support neither inference nor rules, a proposed solution is to translate archetype definitions to an ontology language as OWL that supports integration with rules, for example, SWRL rules (both languages are described in the Background Chapter). The integration with OWL and SWRL gives the archetype paradigm the capacity to reach many systems and development tools that are already designed for the Semantic Web technologies (Golbreich & Imai, 2004). Besides, this research is oriented to evaluate the description logic capabilities offered by OWL when expressing the structural and semantic constraints contained in clinical archetypes.

Going a step beyond, this thesis recommends a flexible approach for reusing declarative knowledge in the form of rules. That form of knowledge is commonly found in clinical guidelines (some examples are provided in section 2.3.2). Then, the representation of the decision points that drive the execution of clinical guidelines can be approached as a complement to existing archetype models. Moreover, attaching rules and reasoning on archetypes can offer consistency checks that help detecting data representation errors and validating archetypes. From a wider perspective it represents a step towards level 3 of Semantic Interoperability, introduced in the last SemanticHEALTH report (V. N. Stroetmann et al., 2009) and described in section 2.1. For example, a reasoner can detect inconsistent restrictions included in some archetype according to the RM or a parent archetype that is being specialised.

One of the greatest advantages of the philosophy of two–level modelling with archetypes resides is allowing the definition and sharing of archetype expressions as a *decentralized process*, that is, a process where large repositories of archetypes are updated and maintained by a variety of cooperating groups of experts or institutions, working on the same or different domains. Following the same philosophy, this research will develop an approach to allow for the collaborative definition of the SWRL rules associated to archetypes.

For example, having the *Blood Pressure* archetype translated to OWL allows any group of experts to define and publish some SWRL rules that detect and evaluate measures' anomalies from a *Hypertension* perspective. At the same time, a different group completely unconnected from the first one can do the same but from a *Hypotension* perspective. As well as an archetype repository, the clinical rule repository will hold the bindings between the SWRL definitions and the OWL version of the archetypes they refer to. It should be noted that SWRL rules can be inserted in the same OWL file as the archetype, as well as in a different one via ontology import. Such Clinical Rule Repository will provide typical repository functionalities, like management, searching, browsing, etc.

In this manner, the goals of present research are oriented to fulfil the semantic interoperability requirements that were pointed out in the European Commission report titled "Semantic Interoperability for Better Health and Safer Healthcare" (V. N. Stroetmann et al., 2009). The report defines semantic interoperability as "the ability, facilitated by ICT applications and systems, to exchange, understand and act on citizens/patients and other health-related information and knowledge among linguistically and culturally disparate health professionals, patients and other actors and organizations within and across health system jurisdictions in a collaborative manner".

1.1.1 Research Objectives

The overall objective of the research presented here is to enhance the archetype approach to clinical information and knowledge representation by means of ontologies and semantic web languages in order to support features that correspond to level 3 of Semantic Interoperability (SIOp) between EHRs (V. N. Stroetmann et al., 2009)**.**

The concrete objectives of the research are:

1. To enable clinical archetypes to be expressed through an ontology language. This includes a deep analysis of archetype constraints in order to adopt the translation methodology that better preserves the archetype's semantics when expressed in an ontology language. The translation will be focused on further supporting integration with rules and inference execution over patient's data as well as binding to existing biomedical terminologies and ontologies.
2. To enable the execution of rules combined with instances of clinical archetypes. Inferential mechanisms will allow reaching new conclusions that expand the boundaries of the declarative knowledge encoded in archetyped data.
3. To define a logical foundation for integrating the archetype model and clinical terminologies models through an ontology context. Properly

binding archetypes to other interoperability related artefacts like terminologies is essential for achieving full SIOp. The proposed mechanism is expected to provide seamless access to terminologies during the inference stage.

1.1.2 Research Contributions

This thesis is aimed at fulfilling the interoperability requirements that clinical archetypes, terminologies and guidelines cannot provide when used in isolation. The main contributions are the following:

- The design and implementation of the ***ArchOnt Framework***, supporting the following features:
    - Full automatic ADL to OWL translation.
    - Support for SWRL rules definition based on previously translated archetypes.
    - Binding of archetypes and subsets of SNOMED-CT through an OWL context.
    - Archetypes instantiation from patient data from relational databases (or from linked data provider).
    - Inference execution over patient data.
- Significant parts of the declarative knowledge contained in the following clinical guidelines will be expressed through the combination of openEHR archetypes and SWRL rules. The integration and encouragement of primary care guidelines, and care pathways in general, has been acknowledged as an essential goal for healthcare interoperability:
    - *Ligation of the Sigmoid or Transverse Sinus during Large Petroclival Meningioma Surgery* (Hwang, Gwak, Paek, D.-G. Kim, & Jung, 2004).
    - *Prevention and Treatment of the Pressure Ulcers* guideline published by NICE[1,2].

[1] NICE - National Institute for Health and Clinical Excellence

[2] http://guidance.nice.org.uk/CG29/

- *Respiratory tract infections - antibiotic prescribing* guideline published by NICE[3].

- The translation algorithm has been implemented and it is currently integrated as a module of the openEHR Java Implementation Project[4]

1.2 Notations & Vocabulary

This section provides a description of the acronyms, abbreviations, terms, notations, etc. frequently used throughout the thesis. It should be noted that some definitions below may depend on each other and many of them will be further explained in the Background Chapter.

- The *Electronic Health Record* will be referred to as **'EHR'**. It is a patient's medical record in digital format. EHR systems coordinate the storage and retrieval of individual records with the aid of computers. A variety of types of healthcare-related information may be stored, processed and accessed in this way.
- The terms **'clinical experts'** and **'experts'** will be used interchangeably to refer to the personnel that is involved in medical research, performing task as archetypes and rules definition, as well as archetype to terminology bindings. Clinical experts typically include practicing doctors, nurses, and clinicians.
- The SNOMED CT terminology may be referred to as **'SNOMED'**. It is essentially a clinical terminology that provides clinical content and expressivity for clinical documentation and reporting. The SNOMED clinical content will be referred to as **'SNOMED concepts/codes'**.
- The *Web Ontology Language* will be referred to as **'OWL'**. It facilitates greater machine interpretability of Web content than that supported by XML and RDF by providing additional vocabulary along with a formal semantics.
- OWL fragments will be provided in *RDF/XML*[5] syntax, as well as in the more concise *Manchester*[6] syntax.

---

[3] http://guidance.nice.org.uk/CG69/
[4] http://www.openehr.org/projects/java.html
[5] http://www.w3.org/TR/rdf-syntax-grammar/

- The term **'translation'** refers to the translation of archetypes expressed in ADL to an equivalent representation in OWL.
- The *Semantic Web Rule Language* will be referred to as **'SWRL'**. It is based on a combination of OWL with the *Rule Markup Language*.
- The phrase **'clinical rules'** refers to propositions of declarative knowledge within the clinical domain, represented by the SWRL language and encoded, for example, in clinical guidelines.
- The term *Interoperability* will be referred to as **'IOp'**, while *Semantic Interoperability* will be referred to as **'SIOp'**. Both terms are explained in the Background Chapter.

openEHR related:

- The *openEHR Reference Model* will be referred to as **'Reference Model/RM'**. It is an information model that defines a logical architecture including a flexible syntax and some generic types of clinical information. The RM is designed to be invariant in the long term, to minimise the need for software and schema updates.
- The *openEHR archetypes* will be referred to as **'clinical archetypes/archetypes'**. An archetype is a computable expression of a domain content model in the form of structured constraint statements, based on the RM, and always expressed in the same formalism.
- The *openEHR Archetype Object Model* will be referred to as **'Archetype Object Model/AOM'**. It specifies the formalism for the definition of archetypes and for the bridge between the RM and knowledge resources. It describes the definitive semantic model of archetypes in the form of an object model.
- The *Archetype Definition Language* will be referred to as **'ADL'**. It is a formal language for expressing archetypes, and can be categorised as a *knowledge description language*. It provides a formal, abstract syntax for describing constraints on a clinical domain entity whose data is described by the RM.
- The *openEHR Archetype profile* will be referred to as **'oAP'**, which defines custom constraint classes for use with the generic AOM.

[6] http://www.w3.org/TR/owl2-manchester-syntax/

- The *openEHR Archetype Model* will be referred to as **'AM'**, defining the structure and semantics of archetypes and templates. The AM consists of the ADL, the AOM and the oAP.
- Elements in the RM and the ADL code will be presented in `Courier font`, whereas the names of the constrained RM elements defined in archetypes will be presented in *italic type* within the text.

1.3 Thesis Outline

Chapter 2 provides a summary of SIOp when applied to the healthcare environment. Goals, trends and requirements for a full SIOp are described. The chapter also provides an overview of used languages and models, introducing ADL, OWL, SWRL, openEHR models in addition to the SNOMED-CT clinical terminology.

Then Chapter 3 gives a review of related work, contrasting existing approaches to archetype translation with the one presented in present research. Some approaches to clinical guidelines execution and bindings between archetypes and clinical terminologies are also analysed.

The main contributions of this research are covered in the next four chapters. Chapter 4 details the principles of the ADL to OWL translation approach. It should be noted that the new features of OWL 2.0 are considered here in order to properly translate some archetype constraints that were abstrusely translated with previous OWL versions. These principles are put into practice in Chapter 5, where the ADL2OWL translator implementation is described. Chapter 6 subsequently reports on a method for attaching reusable clinical rules to the OWL representation of archetypes, providing case studies that are useful for implementing patient safety mechanisms. Another field of application of expressing archetypes through OWL is that of terminology integration and bindings to existing clinical ontologies. Such subject is covered in Chapter 7.

Chapter 8 addresses the evaluation of the ADL2OWL translator as well as the proposed rule system. Finally, Chapter 9 concludes the thesis with an outline of the research performed in the preceding chapters. It re-examines the contributions and objectives pointed out in the introductory chapter and discusses whether the claims have been successfully accomplished. Chapter 11 provides suggestions for future work to take the research forward.

2 Background

Since the beginning of the work on health informatics, one of the most important factors has been the quality of the data. There have been several efforts and studies directed to encourage clinical professionals to use computer systems to distribute and collect patient data, for example, Collen, Van Brunt and Davis (1976), Rogne (1984) and Kuhn et al. (1982). Nevertheless, most of the quality of the data from these researches has been below the expected standard of completeness and consistency, needing additional efforts to improve health systems.

Then an important advance in the field of health informatics was carried out by the health services in the UK. In 2000 the Department of Health UK published the "Information for Health" document to analyze the issues of developing a strategy for Information Management and Technology at the local level of the National Health Service (NHS) (Preston, 2000). This led to further development of electronic health records (EHRs), which has then created a growing need to integrate the various EHRs.

The fact that health care is near a decade behind many other high-risk industries in its attention to ensuring basic safety is partly due to the lack of a single designated government agency devoted to improving and monitoring safety throughout the health care delivery system. In addition, it is due to the existence of several approaches to the same problem, none of which has been yet accepted as a worldwide standard. According to Rector (1999), the problems of standardising the medical language and terminology have been a major concern in health informatics efforts for over a decade. Another reason for this backwardness is that health care data has a number of distinctive characteristics that differentiate it from other industries. The research analysis by Pedersen and Jensen (1998) includes a comparison showing that conventional data warehouses are more complex in general than EHR, see Figure 1.

	Clinical	**Conventional**
Data Model	Simple	Complex
Temporal Support	Medium	Advanced
Classifications	Simple	Advanced
Continuously Valued Data	No	Yes
Dimensionally Reduced	No	Yes
Very Complex Data	No	Yes
Advanced Business Rules	Maybe	Yes(Protocols)
Data Mining	Maybe	Yes(Medical Research)

Figure 1. Conventional versus Clinical Data warehouses. Taken from Pedersen and Jensen (1998).

In addition to the quality of the data, it is also challenging to get doctors to use knowledge-based systems as an aid to the process of data collection and usage. The main reason for the lack of a widespread use of knowledge systems for decision making and data collection, especially in hospitals, is the severe time pressure under which clinicians work. The success of a system is only possible if many people agree to use it for the purposes for which they are conceived. Besides, financial pressures, particularly in secondary care, often lead to cut government budgets on hardware and software. Subsequently, this leads to breakdowns in consistently seeking for the computerization of the health care process (Rector, 1989). Some of the limitations have gradually declined over the last 20 years and there is increasing computerization of the systems of health care. Nevertheless, the need to improve the quality and accuracy of "data" remains alive.

The following sections in this chapter describe the role that clinical data models, ontologies and terminologies play in reaching such quality and accuracy in the healthcare environment. It is also explained why semantic interoperability has become a key goal to achieve an adequate functionality of healthcare systems.

2.1 Interoperability in health systems

The term “semantic interoperability” (SIOp) has been used for more than a decade in computer science as the ability to exchange services and data between components of large-scale, distributed systems in way that ensures the requesters and providers to have a common understanding of the meanings of the requested services and data (Heiler, 1995). With regard to the medical domain, SIOp is mainly associated to the speed and consistency of accessing meaningful health related data. The term has rapidly gained importance since researches like Kalra, Blobel, and Regensburg (2007) and Aspden (2004) pointed out that SIOp is essential if we aim to enable electronic health record information to be shared seamlessly and meaningfully, and computational services are to be able to interpret safely clinical data that has been collected from different sources.

Then a milestone was reached when the “Semantic interoperability for better health and safer healthcare” report was published by the SemanticHEALTH project (V. N. Stroetmann et al., 2009). It is a comprehensive report including the goals, trends and technical challenges to achieve SIOp between heterogeneous healthcare systems. A definition of the term according to this specific domain is provided: “the ability, facilitated by ICT applications and systems, to exchange, understand and act on citizens/patients and other health-related information and knowledge among linguistically and culturally disparate health professionals, patients and other actors and organizations within and across health system jurisdictions in a collaborative manner”.

The areas within healthcare which benefit most from SIOp can be categorized as follows:

- **Patient care:** The benefits in this area include medical staff saving work time, gaining efficiency and improving safety and clinical outcomes through better access to patient information across disciplines, care settings and countries. These influence *patient safety*, *dissemination of good practice*, *integration of education and care*, *connecting multiple locations for collaborative care delivery* and *empowerment of citizens*, among others.

- **Public health:** Benefits are also associated to being able to use richer clinical detail, leading to improvement and greater confidence in information used for audit, planning and performance management. These influence *international statistics*, *comparative outcome assessment*, *pharmacovigilance*, *coordination of risk assessment*, *management and surveillance of large-scale adverse health events* and *population health research*, among others.
- **Research and translational medicine:** SIOp achievement can also lead to the development of *multi-centre studies and trials*, *health data repositories*, *bio- and tissue-banks* and *personalised medicine based on genetic and genomic analyses*, among others.
- **Support for diverse markets:** SIOp provides means for the *identification of solutions with superior benefit/cost ratios*, *enabling plug and play best of breed*, *encouraging industry involvement*, *stimulating innovations by health service providers and involving clinicians* and *harmonising legal and regulatory frameworks*.

SIOp not only offers means for new methods and services in the health domain but also contributes to maximise all the benefits from the relation between medicine/healthcare and information technology, given the constraints in resources. These efficiency gains have been emphasized by studies such as Walker et al. (2005) that estimates that US$77.8 billion per year could be saved by implementing fully standardized electronic health care information exchange and interoperability (HIEI) between providers and independent laboratories, radiology centres, pharmacies, payers, public health departments and other providers in the US. Another study from the RAND Corporation (Girosi, Meili, & Scoville, 2005) considered that standardised information exchange systems could result in net savings of as much as 5% of current US healthcare expenditure.

As interoperability is not a binary variable but rather a scale reaching from zero to full IOp, it should be noted that the benefits listed above can be achieved to a greater or lesser extent depending on the degree of level of interoperability present. The definition of such levels has evolved from a general perspective oriented to any kind of information system (Sheth, 1999) to a more specific definition in accordance to the requirements of a particular domain. In this manner, Garde et al. (2007)

provided three levels adapted to the healthcare environment. Two years later, the levels of IOp established by the SemanticHEALTH report (V. N. Stroetmann et al., 2009) were the following:

- **Level 0**: no interoperability at all.
- **Level 1**: technical and syntactical interoperability (no semantic interoperability)
- **Level 2**: two orthogonal levels of partial semantic interoperability
  - *Level 2a*: unidirectional semantic interoperability
  - *Level 2b*: bidirectional semantic interoperability of meaningful fragments
- **Level 3**: full semantic interoperability, sharable context, seamless co-operability, that will allow gaining the benefits of computerized support for reminders, alerts, decision support, workflow management and evidence based healthcare, i.e. to improve effectiveness and reduce clinical risks. In Level 3 the use of an EHR reference model, a rich library of clinical data structures, and the definitions of terminology bindings to value lists for each element of the data structures have all to be agreed within a record sharing community.

However, reaching high levels of interoperability is a resource intensive task while nowadays it is difficult to associate the benefits of interoperability with those who pay for it. Therefore, one of the objectives of this research is to provide a convincing demonstration of the benefit of migration from ad-hoc to interoperable systems.

For the purpose of increasing the level of IOp and attain full SIOp, certain technological trends and challenges have been identified by the SemanticHEALTH report. They revolve around some key information and knowledge artefacts such as ***clinical archetypes***, ***ontologies***, ***terminologies*** and ***rules***[7]. In general, these artefacts must be combined, integrated and enhanced in order to ensure precision of meaning, consistency, understandability and reproducibility of clinical data and information. Narrowing down to more precise recommendations, the following list contains the ones which are directly encouraged by this thesis:

[7] These concepts are introduced in further sections of the Background.

- Adopt a standardised approach for representing and sharing of clinical data structure specifications: agree to use archetypes.
- Develop and integrate ontologies into decision support and knowledge management software.
- Collaborate on key use cases for shared care and patient safety, and on defining and tidying the corresponding SNOMED CT subsets.
- Seed clinical forums to develop care pathways and archetypes to meet the needs of safe and evidence based care in different medical domains and disciplines.
- Enable guidelines, care pathways, alerting/monitoring and decision support components to function effectively and safely across heterogeneous systems.
- Archetype indexing, ontology binding to archetypes, and archetype/template repository services.
- Development of methodologies and tools for binding of archetypes and HL7 v3 messages to terminologies and ontologies.
- The use of SNOMED CT with archetypes and HL7 v3.
- Research on ontology driven architectures for clinical medicine.

There are many different and independent subareas inside the healthcare field where these advances could be applied to get the benefits from Level 3 of SIOp. However, it is recognised that achieving the highest level across the entirety of healthcare would be a lengthy, expensive and possibly unattainable goal. It is instead recommend by the SemanticHEALTH report to search for full SIOp in specific subareas of clinical practice with high patient safety risk, and in priority subareas for which the evidence is strongest for a gap to be bridged between current and good practice. **Listed below are three of those specific subareas which are addressed by this thesis:**

- New medication prescriptions requiring comprehensive information on concurrent medication and details of known allergies and conditions.
- Reminders and prompts for overdue or overlooked health care actions and interventions.

- Evidence-based care, the use of clinical guidelines and other forms of evidence to determine the optimal management strategy and care pathway for a given patient.

2.2 The two-level paradigm and the openEHR approach

For decades, patient data was stored in the form of free text narratives in clinical notes, or in prescription notes. As this kind of documents are abundant in the healthcare domain and they need to be mapped to specific terminologies, there are projects such as the Medical Subject Headings (MeSH) that work with clinical narratives to extract relevant information for storage in EHRs or for coding to standard terminologies (Lowe & Barnett, 1994). However, poor and incomplete quality data that may lead to negligence and medical errors has brought about awareness to improve the methods employed to record data. Besides, the field is difficult due to lack of context. It is difficult to trust the results without knowing the context in which a particular clinical statement was carried out. In recognition of the importance of context, efforts are being made to increase the utilization of structured data entry, in the form of electronic health records (EHRs), which can control the quality and context of data. EHRs also provide sufficient traceability and reduction of omissions and errors.

Thus, EHRs have been developed mainly to address the limitations of most existing systems to record the fine grained elements in a structured way, which are required for health care and clinical decision making (Rector, Nowlan, Kay, Goble, & Howkins, 1993). Coiera (1997) asserts that if the data contained in EHRs are to be analysed they need to be accessible in some regular way.

It should be also noticed that healthcare systems development have classically followed similar steps to those of other IT domains. That is, requirements are gathered via ad-hoc discussions with users (typically based on the well-known "use case" methodology), designs and models built from the requirements, implementation proceeds from the design, followed by testing and deployment and ultimately the maintenance part of the lifecycle. This procedure is usually characterised by ongoing high costs of implementation change and a widening gap between system capabilities and the requirements at any moment. The approach also suffers from the fact that ad-hoc conversations with systems users frequently fail to reveal underlying content and workflow. Besides, the collaboration is rarely effective between the main two groups of professionals interacting in this domain, i.e. information science experts and health professionals. Without such collaboration it is not possible to achieve any efforts at developing safer health information systems that interoperate seamlessly at different levels of granularity. On one hand, clinical experts may not be well-versed in the field of information technology, thus being unaware of the technical limitations of certain solutions proposed by them. On the other hand, information science experts do not have the clinical background and expertise to independently develop systems that meet the requirements of the health professionals, and the simplicity and understandability ones. Further problems with the classical modelling approach are pointed out by Beale (2002).

A promising solution to these inconveniences is to model the clinical domain using the so called two-level modelling approach (J. Grimson et al., 1998; Beale, 2002; Garde et al., 2007). The openEHR Foundation[8] is an international organization comprising a non-for-profit company and an online community supporting the development of specifications and tools for EHR interoperability according to the archetype methodology and the two-level approach. Another initiative supporting archetypes is the European Committee for Standardization (CEN)[9], Technical Committee 251 (CEN/TC 251)[10], that has produced CEN 13606, an electronic health record (EHR) extract standard based on archetypes. Currently the standard is also approved by ISO[11] so it is known as CEN/ISO 13606. This thesis is based on openEHR instead of CEN/ISO 13606 since the first can be considered a superset of the latter (Schloeffel, Beale, Hayworth, Heard, & Leslie, 2006) thus providing richer built-in semantics.

According to the openEHR approach, the two-level model contains a generic RM that defines a logical information architecture for the interoperability of EHR systems, constituting a base representation framework. The RM represents the

[8] http://www.openehr.org
[9] http://www.cen.eu
[10] http://www.cen.eu/cen/Sectors/Sectors/ISSS/Committees
[11] http://www.iso.org

general features of the components of the EHR, how they are organized and the context information needed to satisfy both the ethical and legal requirements of the record. It includes a flexible syntax and some generic types of clinical information as *observations*, *evaluations*, *instructions* and *actions*. Then, instances or specialisations of that RM are devised in the form of constraints expressed through more concrete "archetypes", which serve as a shared language for common and specialised clinical concepts. In other words, the RM encloses the stable features like the set of classes that make up the blocks constituting an EHR and the basic syntax of statements, while archetypes allow for sharing a wide variety of combinations of those classes corresponding to EHR fragments created for specific clinical situations. For example, "blood pressure", "medication order" and "transfusion" are clinical statements that have already been specified as archetypes, so they can be used or refined as reference data structures for the interchange of clinical data.

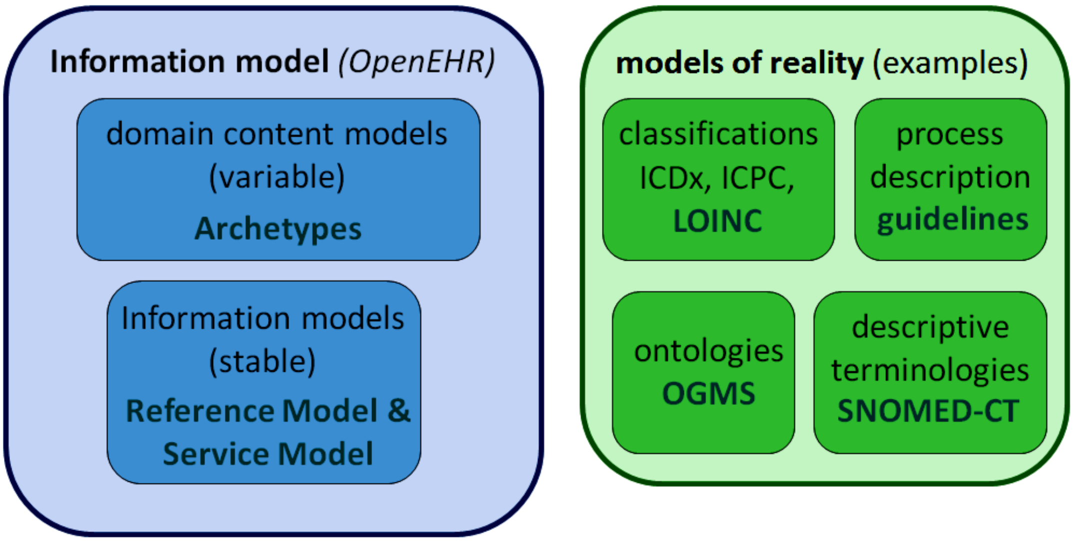

Figure 2. Separation of models of information and models of reality. Adapted from Beale and Heard (2008a).

Predecessors of the current openEHR archetype approach were first proposed and developed across Europe in 1996-1998, through the *Synapses Project* (W. Grimson et al., 1998) and in Australia in 1997-1999 within the *Good Electronic Health Record Project* (Beale, 1999; Heard, 2000). Both approaches stated that generic information models to represent EHR data gave considerable freedom to clinicians and implementers. They could define hierarchical representations of specific clinical record entries in potentially different ways. These archetypes ancestors sought to standardise the way in which concepts should be represented in the clinical health record within generic EHR models, to fulfil clinical requirements and guarantee interoperability.

As said before, the OpenEHR Model is separated in two layers, a variable one described by AM and a stable one described by the RM. Both of these models are different in the levels of abstraction from the models of reality such as classifications and ontologies which are describing the real phenomena. Such separation is illustrated in Figure 2. However, in spite of the fact that they have different types of authors, representations and purposes, they are complementary knowledge artefacts that can and should be integrated, in order to allow for SIOp and improve patient safety.

One of the greatest advantages of the philosophy of two–level modelling with archetypes resides in that it allows the definition and sharing of archetype expressions as a *decentralized process*, that is, a process where large repositories of archetypes are updated and maintained by a variety of cooperating groups of experts or institutions, working on the same or different domains. An example of such repositories is the one provided by openEHR (see Figure 7).

Nevertheless, as this approach is becoming widely accepted, it is certain that the number of available archetypes will become very large and hard to manage. Besides, such decentralized development of archetype definitions is exposed to content overlapping and limitations in the normalization scope. The translation approach proposed in this thesis encourages the development of better integration and management environments for archetypes by supporting semantic metadata and classification.

2.2.1 The openEHR Reference Model, Archetypes and Templates

The formal definition provided by openEHR (Beale & Heard, 2007) is that an archetype is a computable expression of a domain content model in the form of structured constraint statements, based on a reference (information) model. OpenEHR archetypes are based on the openEHR RM. Archetypes are all expressed in the same formalism. In general, they are defined for wide reuse; however, they can be specialised to include local particularities. They can accommodate any number of natural languages and terminologies.

Level of Interoperability	**Main mechanism for interoperability**	**Description**
Syntactic (data) interoperability	openEHR reference model (RM)	The openEHR RM alone ensures syntactic (data) interoperability independent of any defined archetypes. The openEHR RM does not define clinical knowledge; this is defined and communicated by archetypes separate from the RM. Hence, data items are communicated between systems only in terms of clearly defined, generic RM instances. As the RM is stable, achieving syntactic interoperability between systems is undemanding.
Structural interoperability	archetypes	Structural interoperability is achieved by the definition and use of archetypes. As agreed models of clinical or other domain specific concepts, archetypes are clinically meaningful entities. An EHR entry which has been archetyped will have the same meaning no matter where or in which EHR it appears. Thus, archetypes can be shared by multiple health systems and authorities, enabling information to be shared between different systems and types of healthcare professionals. Clinical knowledge can be shared and clinical information can be safely interpreted by exchanging archetypes.
Semantic interoperability	Domain Knowledge Governance	The use of archetypes and the RM alone do not guarantee that different EHR systems and vendors will construct equivalent EHR extracts, and use the record hierarchy and terminology in consistent ways. Thus, this alone does not ensure semantically interoperable systems. For semantically interoperable systems, archetype development must be coordinated through systematic "Domain Knowledge Governance" to, for example, avoid incompatible, overlapping archetypes for essentially the same concept.

Figure 3. The openEHR approach to interoperability and the two-level model shaded in gray. Taken from Garde et al. (2007).

As an example of the two-level modelling approach consider the *Systolic Blood Pressure (SBP)* included in the *Blood Pressure* archetype. The SBP is a clinical concept defined at the archetype level and representing the peak systemic arterial blood pressure of a patient in a given moment. However, its definition is a specialisation of the `DV_QUANTITY` class that belongs to the RM level. In general, `DV_QUANTITY` instances can store any magnitude and its unit, among other attributes, but in the *SBP* specialisation, the magnitude is constrained to the values less than 1000 and not less than cero, measured in millimetres of mercury (*mmHg*) as the mandatory unit. The *Blood Pressure* archetype is in turn a specialisation of a wider RM class `OBSERVATION`.

An archetype gives specifications of pre-defined constraints on the data recorded and also of the data structure including multiplicity, optionality and relevant bindings to terminology systems and natural language. An archetype definition can specialise or contain other archetypes, and can reuse or import blocks of elements occurring previously in the same archetype or in another archetype. Archetype specialisation allows including local peculiarities. For example, the *Examination of the fetus* archetype and the *Examination of the uterus* archetype are both specialisations of the *Examination* archetype, according to the openEHR Clinical Knowledge Manager[12].

As IOp is one of the main goals of the openEHR approach, it is important to understand the impact of the two-level model on such feature. The table in Figure 3 describes the influence of each level on the IOp of healthcare systems.

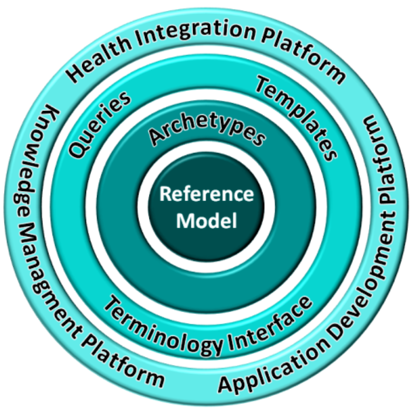

Figure 4. The openEHR Health Computing Platform. Adapted from Beale and Heard (2008a).

While archetypes are generally broad models, that have very open compositional possibilities, there is a need to narrow the choices of archetypes for specific or local purposes. *Templates* are used to accomplish this task. They can control the following:

[12] http://openehr.org/knowledge/

- archetype composition, or chaining
- reduction in allowed terms
- restricting optionality
- removing structures defined in the referenced archetypes

For instance, a hypertension recording may include fragments from a blood pressure archetype, a cholesterol archetype and a drug medication archetype. The formal definition by Beale and Heard (2007) states that a *template* is a directly locally usable definition which composes archetypes into larger structures often corresponding to a screen form, document, report or message. Templates may add further local constraints on the archetypes it mentions, including removing or mandating optional sections, and may define default values.

Figure 4 is extracted from Beale and Heard (2008a) and summarizes the openEHR Health Computing Platform where archetypes are combined together through templates, and used at runtime to extract data, to enable querying, and to support legacy data transformation.

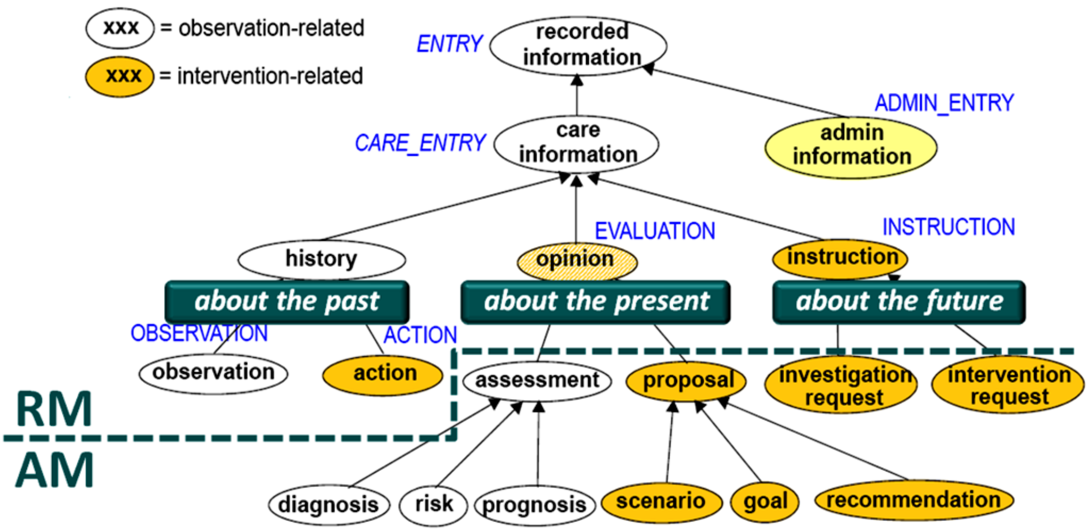

Figure 5. The boundary between the RM and the AM. Adapted from Beale and Heard (2008a).

Then, according to the purpose of clinical information, the openEHR Reference Model (RM) makes the classification in Figure 5. The dotted line represents the frontier between the RM and the AM. As the components in the latter are free to be developed according to particular necessities of third party institutions, the concepts shown in the AM area are just a recommendation. In this manner, every piece of recorded information will be an instance of an archetype in the area below the dotted line in Figure 5. In turn, every archetype will be considered as a specialisation of the RM, which is represented above the dotted line.

The RM states that all information created in the clinical statement context will be expressed in terms of `ENTRY` instances in the Information Model. Thus, archetypes inherit the attributes and properties of not only the immediate entity that they specialise but also their parent entities. The `ENTRY` class in the openEHR RM has four subclasses: `OBSERVATION`, `ACTION`, `EVALUATION` and `INSTRUCTION`. For example, besides the two `OBSERVATION` attributes (i.e. `data` and `state`), the subclass also inherits `ENTRY` attributes such as `subject`, and `protocol`. The `cardinality` constraints are also inherited along with the attributes. Therefore, an instance of `OBSERVATION` can only have one `subject` and an optional `protocol`, as defined in the `ENTRY` package in Figure 6.

With regard to clinicians' perspective, `OBSERVATIONs` and `ACTIONs` are used to record information about the past while `EVALUATIONs` are about the present and `INSTRUCTIONs` describe events to be carried out in the future.

As described by Beale, Heard, Kalra, and Lloyd (2008), `OBSERVATIONs` are distinguished from `ACTIONs` in that `ACTIONs` record interventions whereas `OBSERVATIONs` record only information relating to the situation of the patient, not what is done to him/her. An `EVALUATION` may say that "*oral cortico-steroids are indicated at a peak flow of 200 l/m*". A corresponding `INSTRUCTION` would be issued to indicate the actual drug, route, dose, frequency, and so on.

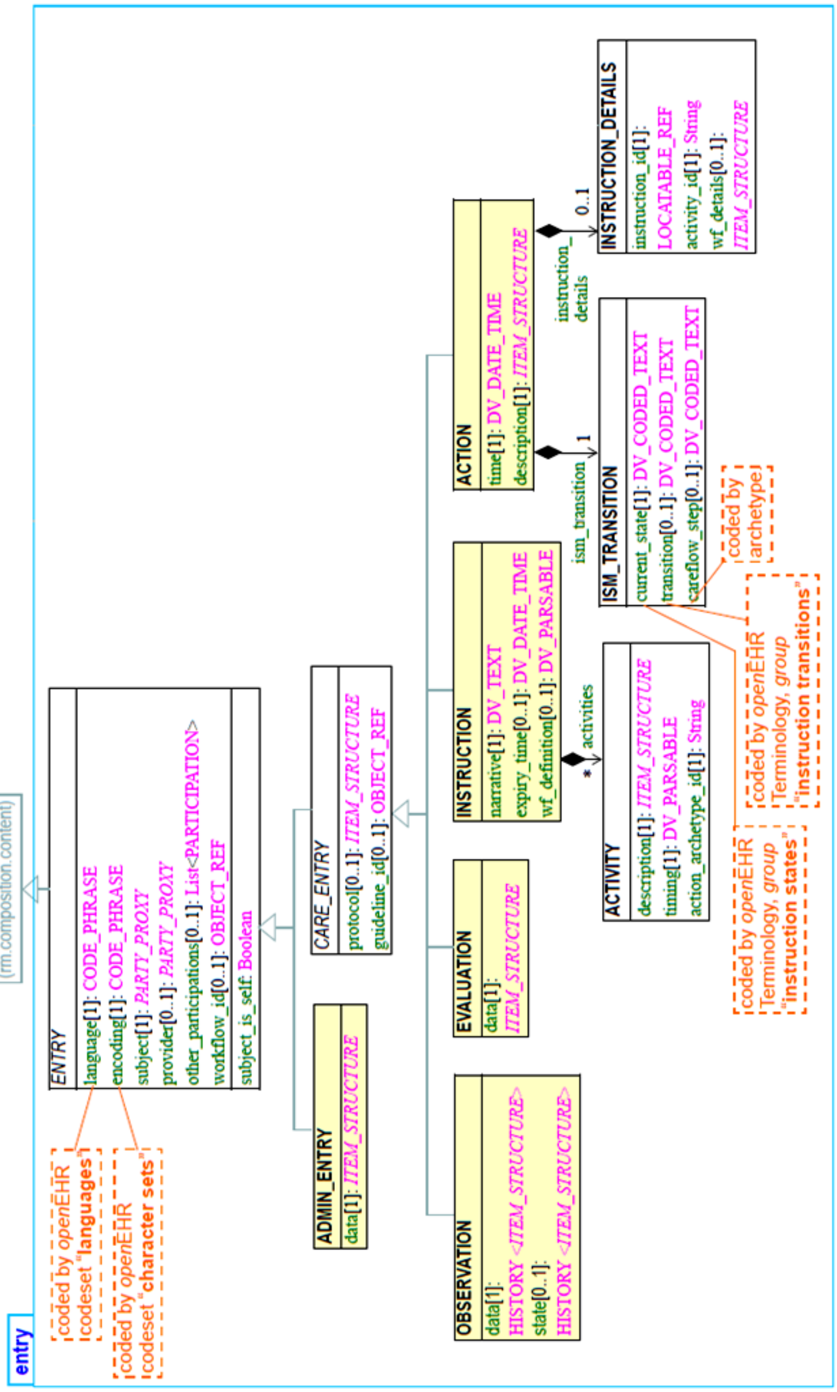

Figure 6. The RM `ENTRY` package as defined by Beale et al. (2008).

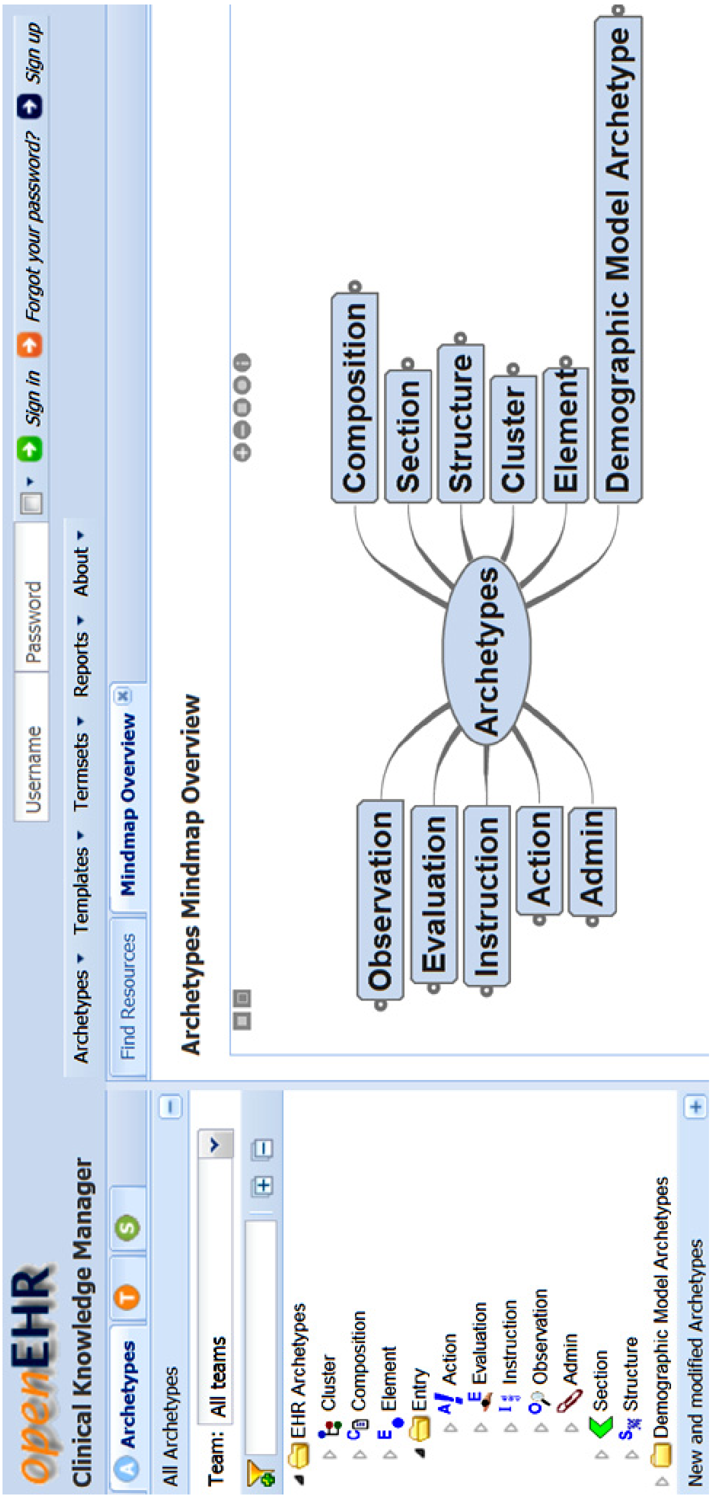

Figure 7. The openEHR Clinical Knowledge Manager[13] (*archetypes repository*) currently contains near 300 archetype definitions.

[13] http://openehr.org/knowledge/

The separation of clinical information in categories or types is essential for achieving semantic interoperability. Take as example that the "high blood pressure" statement can be related to a finding (`OBSERVATION`), but can also be a diagnosis of *hypertension* disease included in an `EVALUATION` entry. There is an important semantic difference that can be represented in this case by using different RM classes.

2.2.2 The Archetype Definition Language and the AOM

The openEHR Archetype Definition Language, or ADL, is a formal language for expressing archetypes that can be categorized as a knowledge description language. It provides a formal, abstract syntax for describing constraints on any domain entity whose data is described by an information model. The syntax is congruent with Frame Logic queries (Kifer, Lausen, & Wu, 1995). It is primarily useful when very generic information models are used for representing all data in a system, for example, where the logical concepts Patient, Doctor and Hospital might all be represented using the class Party, Address, and related generic classes. Archetypes are then used to constrain the valid structures of instances of these generic classes to represent the desired domain concepts. In this way, future-proof information systems can be built and relatively simple information models and database schemas can be defined while archetypes supply the specific modelling, completely outside the software.

An ADL file starts with a header section followed by a definition section and an ontology section (Beale & Heard, 2008b). ADL is divided into two main syntaxes: the data definition syntax or *dADL*, and the constraints definition syntax or *cADL*. The header and ontology sections of the archetype are written in *dADL* and the main definition section of an ADL archetype is written in *cADL*, containing constraints on the data. The header section uniquely identifies the archetype and the clinical concept involved, and includes metadata about the archetype (e.g. its purpose and use). The definition section contains constraints in a tree-like structure created from the reference information model. Finally, codes representing the meanings of nodes and constraints on text or terms as well as bindings to terminologies such as SNOMED-CT (see section 2.4), are stated in the ontology section of the archetype. However, these are optional and they are not available in most of the archetypes openly published on the Web nowadays. Archetypes should fulfil a set of design principles[14], like for example that an archetype can be a specialisation of another archetype and they must be neutral with respect to terminologies. Fragments from a typical ADL file[15] are shown below:

```
archetype (adl_version=1.4)
    openEHR-EHR-OBSERVATION.heart_rate-pulse.v1
specialise
    openEHR-EHR-OBSERVATION.heart_rate.v1
concept
    [at0000.1]      -- Pulse
language
    original_language = <[ISO_639-1::en]>
description
    original_author = <
            ["name"] = <"Sam Heard">
            ["organisation"] = <"Ocean Informatics">
            ["email"] = <"sam.heard@oceaninformatics.com">
            ["date"] = <"26/03/2006">
    >
    details = <
      ["en"] = <
        language = <[ISO_639-1::en]>
          purpose = <"To record the mechanical rate, rhythm and character of the
                    pulse as evidence of 'out-put' heart rate.">
          use = <"For recording mechanical rate of the heart as determined by palpation or
                suitable device.">
          keywords = <"rate", "rhythm", "character", "pulse">
              misuse = <"Not for recording peripheral pulses.">
          copyright = <"copyright (c) 2010 openEHR Foundation">
      >
    >
    lifecycle_state = <"Initial">
    other_contributors = <>
    other_details = <
            ["MD5-CAM-1.0.1"] = <"555A747F3BEA5BCB86F63A0D5C003BEA">
            ["references"] = <"">
    >
```

Figure 8. ADL header section.

```
definition
  OBSERVATION[at0000.1] matches { -- Pulse
    data matches {
      HISTORY[at0002] matches {  -- history
        events cardinality matches {1..*; unordered} matches {
          EVENT[at0003] occurrences matches {0..*} matches {    -- Any event
            data matches {
              ITEM_TREE[at0001] matches {      -- structure
                items cardinality matches {0..*; unordered} matches {
                  ELEMENT[at1005.1] occurrences matches {0..1} matches { -- Pulse present
                    value matches {
                      DV_BOOLEAN matches {
                        value matches {True, False}
                      }
                    }
                  }
                  ELEMENT[at0004] occurrences matches {0..1} matches {  -- Rate
                    value matches {
                      C_DV_QUANTITY <
                        property = <[openehr::382]>
                        list = <
                          ["1"] = <
                              units = <"/min">
                              magnitude = <|>=0.0|>
                              precision = <|0|>
                          >
                        >
```

[14] http://www.openehr.org/releases/1.0.2/architecture/am/archetype_principles.pdf

[15] Most of the ADL files for the archetypes cited in this thesis are available at http://openehr.org/knowledge/. The rest has been developed locally in collaboration with clinical experts from the Fuenlabrada Hospital and the Henares Hospital.

```
                    >
                  }
                }
            ...
            ...
                ELEMENT[at0005] occurrences matches {0..1} matches {  -- Rhythm pattern
                  value matches {
                    DV_CODED_TEXT matches {
                      defining_code matches {
                        [local::
                        at0006,     -- Regular
                        at0007,     -- Regularly Irregular
                        at0008]     -- Irregularly irregular
                      }
                    }
                  }
                }
                ELEMENT[at0.11] occurrences matches {0..1} matches {-- Missed beats/minute
                  value matches {
                    DV_COUNT matches {
                      magnitude matches {|>=0|}
                    }
                  }
                }
        ...
    ...
```

Figure 9. ADL definition section (*data constraints*).

```
        state matches {
         ITEM_TREE[at0012] matches {      -- List
          items cardinality matches {0..*; unordered} matches {
            ELEMENT[at0013] occurrences matches {0..1} matches {   -- Position
             value matches {
               DV_CODED_TEXT matches {
                defining_code matches {
                  [local::
                  at1000,        -- Lying
                  at1001,        -- Sitting
                  at1002,        -- Reclining
                  at1003;        -- Standing
                  at1001]        -- assumed value
                }
              }
            }
          }
        ...
    ...
```

Figure 10. ADL definition section (*state constraints*).

```
                protocol matches {
                  ITEM_TREE[at0010] matches {     -- List
                    items cardinality matches {0..*; unordered} matches {
                      ELEMENT[at1019] occurrences matches {0..1} matches { -- Method
                        value matches {
                          DV_CODED_TEXT matches {
                            defining_code matches {
                              [local::
                              at1020,    -- Auscultation
                              at1021]    -- Device
                            }
                          }
                        }
                      }
          ...
                      allow_archetype CLUSTER[at1013]occurrences matches{0..1} matches {--Device
                        include
                          archetype_id/value matches
                            {/openEHR-EHR-CLUSTER\.device(-[a-zA-Z0-9_]+)*\.v1/}
                        exclude
                           archetype_id/value matches {/.*/}
                      }
          ...
    ...
```

Figure 11. ADL definition section (*protocol constraints*).

```
ontology
    terminologies_available = <"SNOMED-CT", ...>
    term_definitions = <
      ["en"] = <
        items = <
          ...
          ...
          ["at1005"] = <
              text = <"Pulse present">
              description = <"The heart rate is present (implied true if rate >0).">
          >
          ["at1006"] = < ...
          ...
    term_bindings = <
      ["SNOMED-CT"] = <
        items = <
           ...
           ...
          ["at0000"] = <[SNOMED-CT::364075005]>
          ["at0000.1"] = <[SNOMED-CT::248627000]>
          ["at0004"] = <[SNOMED-CT(2009)::78564009]>
            ...
        >
      >
    >
```

Figure 12. ADL ontology section.

Archetypes are themselves instances of the openEHR Archetype Object Model (AOM) that specifies the formalism for their definition. The AOM is a model of the semantics of archetypes that defines an equivalent object model in terms of a UML[16] model. Within such model, the archetype definition is an instance of the `C_COMPLEX_OBJECT` class, which is the root of the constraint structure of an archetype, depicted in Figure 13. The last section of an archetype, the ontology, is represented by its own class, and is what allows the archetypes to be natural language-and terminology-neutral.

Therefore, the AOM is a generic model, meaning that it can be used to express archetypes for any reference model in a standard way. The ADL and the AOM are supported by an ADL parser that can read ADL archetype texts, and whose parse tree results in memory object representation. A parser implementation to handle ADL archetypes is provided by the openEHR Java Implementation Project (R. Chen & Klein, 2007). The ADL to OWL translation implementation described in Chapter 4, which is Java based, is been integrated as a module of that openEHR project[17,18].

---

16 http://www.uml.org/
17 http://www.openehr.org/projects/java.html
18 http://www.openehr.org/svn/ref_impl_java/SANDBOX/ehr2ont/

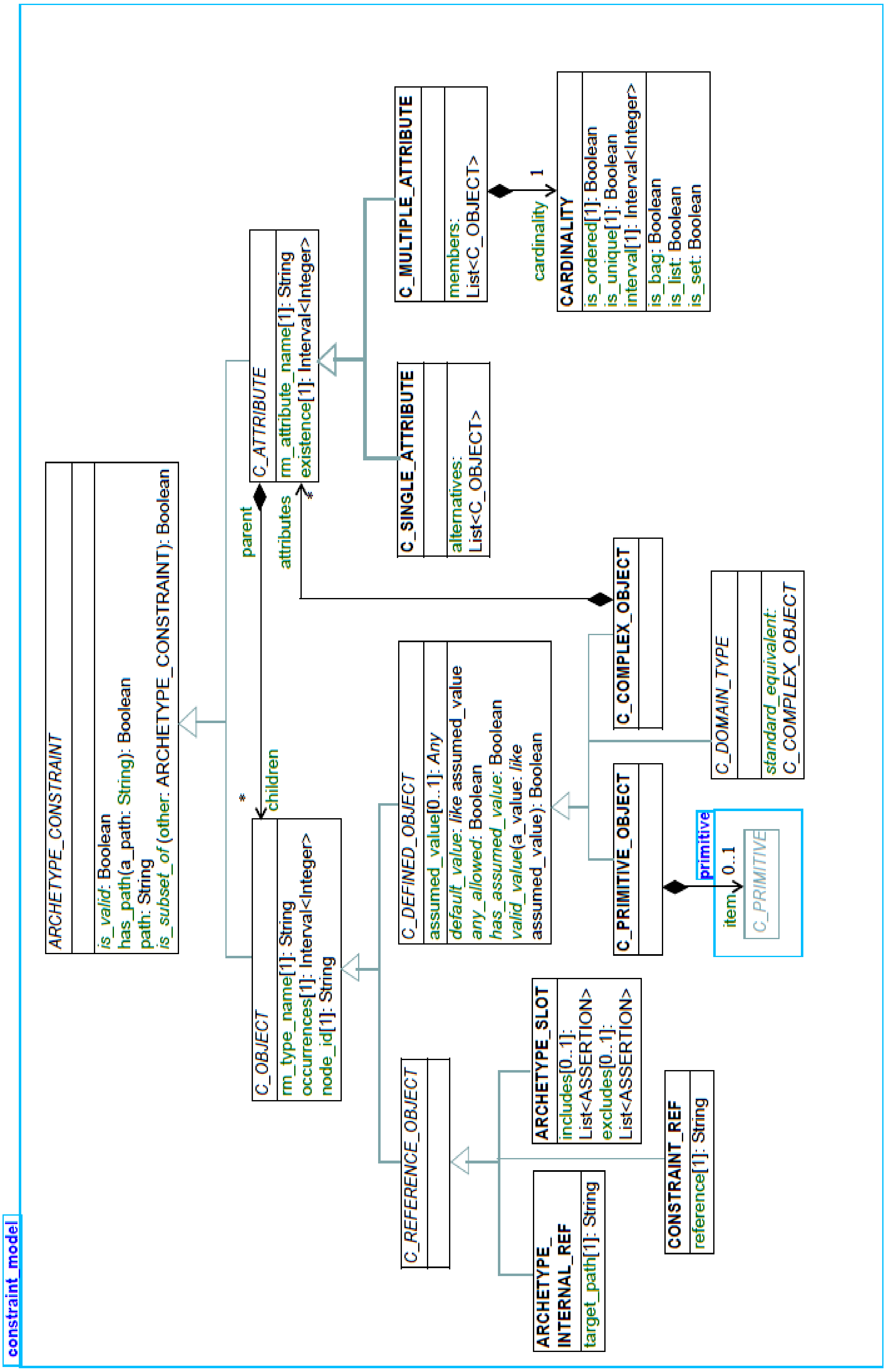

Figure 13. The `Constraint_model` package as defined by Beale (2007).

2.3 Semantic Web Ontologies and Rules

There is an evolving extension of the World Wide Web known as the *Semantic Web*[19] (Berners-Lee, Hendler, & Lassila, 2001) which provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by the W3C with participation from a large number of researchers and industrial partners. "The Semantic Web is an extension of the current Web in which information is given a well defined meaning, better enabling computers and people to work in cooperation" (Guha, McCool, & Miller, 2003).

The Semantic Web is envisioned as the next generation of the WWW, according to Christiaens (2006), it is a Web in which all content has machine-processable meaning. This Semantic Web provides all the functionality needed to build the Pragmatic Web on top of it. Communities will no longer search, but rather find and use information in this Pragmatic Web. The explicit meaning, understandable by both human and machine agents, attached to content is necessary for proper information retrieval and usage.

Ontologies are considered one of the pillars of the Semantic Web. The technical term was introduced in computer science by Gruber (1993). In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes, attributes and relationships. The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application (Gruber, 2009). Common components of ontologies include:

- *Individuals*: instances or objects (the basic or "ground level" objects)
- *Classes*: sets, collections, concepts, types of objects, or kinds of things.
- *Attributes*: aspects, properties, features, characteristics, or parameters that objects (and classes) can have
- *Relations*: ways in which classes and individuals can be related to one another

[19] http://semanticweb.org

- *Function terms*: complex structures formed from certain relations that can be used in place of an individual term in a statement
- *Restrictions*: formally stated descriptions of what must be true in order for some assertion to be accepted as input
- *Rules*: statements in the form of an if-then (antecedent-consequent) 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. This definition differs from that of "axioms" in generative grammar and formal logic. In these disciplines, axioms include only statements asserted as a priori knowledge. As used here, "axioms" also include the theory derived from axiomatic statements.

Ontologies are used in artificial intelligence, the Semantic Web, software engineering, library science and information architecture as a form of knowledge representation about the world or some part of it. It is been several years since ontologies began to be used for representing biomedical knowledge. For example, researches such as Schulz and Hahn (2005) and Smith (2006) formalized medical concepts by means of ontologies. One of the most important advances in bioinformatics was the development of the Gene Ontology (Ashburner et al., 2000), which popularized the development of bio-ontologies. Currently there are large projects and consortiums like the OBO Foundry[20] ensuring the coordinated development of these ontologies and also the United States have established research centres in biomedical ontologies such as the National Center for Biomedical Ontology[21]. There are other previous works that make use of ontologies for tasks related to electronic medical records management (J. S. Rose et al., 2001; Nardon & Moura, 2004; Smith & Ceusters, 2005).

Ontologies are usually accompanied by some document in a formal ontology language. With regard to the semantic web, there have been several approaches (Pulido et al., 2006). Among those, the Ontology Web Language (OWL)[22] is a W3C recommendation for an ontology description language that has gained widespread

[20] http: //www.obofoundry.org
[21] http://bioontology.org
[22] http://www.w3.org/TR/owl-overview/

adoption and for which a considerable number of tools have been developed. It has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

2.3.1 The Protégé-OWL API

Protégé[23] is a flexible, configurable platform for the development of arbitrary model-driven applications and components. It has an open architecture that allows programmers to integrate plug-ins, which can appear as separate tabs, specific user interface components (widgets), or perform any other task on the current model. The Protégé-OWL editor provides many editing and browsing facilities for OWL models, and therefore can serve as an attractive starting point for rapid application development. Developers can initially wrap their components into a Protégé tab widget and later extract them to distribute them as part of a stand-alone application.

The Protégé-OWL API[24] is an open-source Java library for OWL and RDF(S). The API provides classes and methods to load and save OWL files, to query and manipulate OWL data models, and to perform reasoning based on Description Logic engines. Furthermore, the API is optimized for the implementation of graphical user interfaces.

Jena[25] is one of the most widely used Java APIs for RDF and OWL, providing services for model representation, parsing, database persistence, querying and some visualization tools. Protégé-OWL API (v 3.4) and lower versions are integrated with Jena. The Jena ARP parser is used in the Protégé-OWL parser and various other services such as species validation and datatype handling have been reused from Jena. This integration allows using certain Jena functions at run-time, without having to go through the slow rebuild process each time. The architecture of this integration is illustrated in Figure 14.

---

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

[24] http://protege.stanford.edu/plugins/owl/api/

[25] http://jena.sourceforge.net/

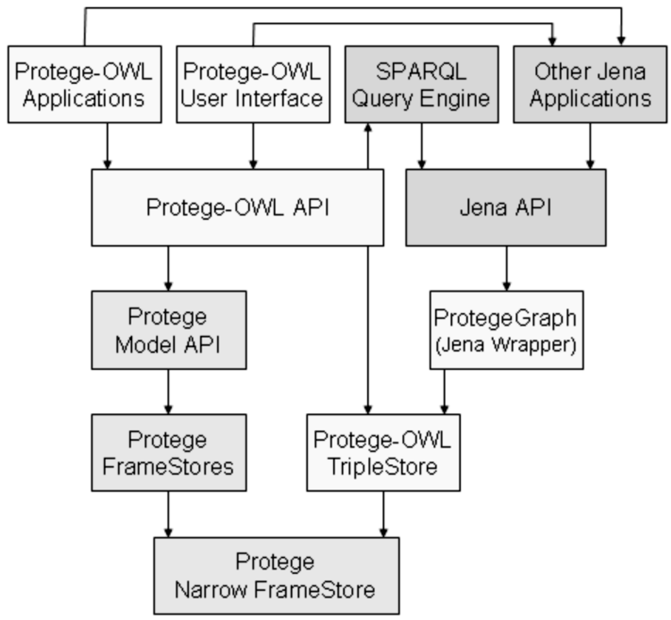

Figure 14. Integration of Jena in Protégé-OWL.

The key to this integration is the fact that both systems operate on a low-level "*triple*" representation of the model. Protégé has its native frame store mechanism, which has been wrapped in Protégé-OWL with the *TripleStore* classes. In the Jena world, the corresponding interfaces are called *Graph* and *Model*. The Protégé *TripleStore* has been wrapped into a *Jena Graph*, so that any read access from the Jena API in fact operates on the Protégé triples. In order to modify these triples, the conventional Protégé-OWL API is used. However, this mechanism allows the use of Jena methods for querying, while the ontology is edited inside Protégé.

The interfaces of the Protégé-OWL model are arranged in an inheritance hierarchy. An overview of the available interfaces can be found in Figure 15, created by Matthew Horridge. The base interface of all resources is *RDFResource*, from which subinterfaces for classes, properties and individuals are derived.

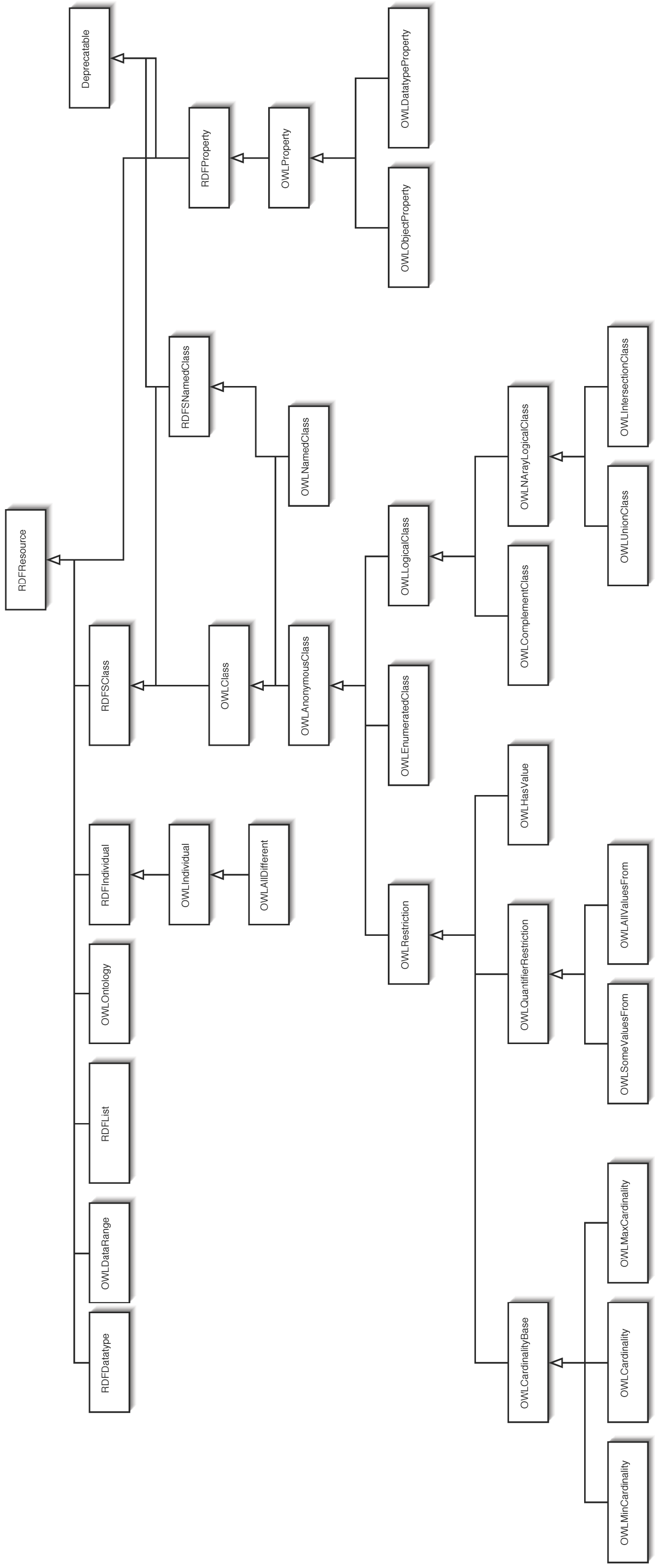

Figure 15. Class diagram of the Protégé-OWL model.

There is a clear distinction in the model between *named* classes and *anonymous* classes. Named classes are used to create individuals, while *anonymous* classes are used to specify logical characteristics (restrictions) of named classes. Logical class definitions can be used to build complex class expressions out of restrictions and named classes. Like restrictions, logical classes are only meaningful if they are attached to a specific named class or property. With regard to the ADL to OWL translation, named classes will represent the RM specialisations while *anonymous* classes like *OWLUnionClass* and *OWLIntersectionClass* will be used to attach the archetype constraints to such specialisations.

The translator implementation that is described in Chapter 5 was originally based on the Protégé 3.4 open source platform that provides both an ontology editor and the Protégé-OWL API (v 3.4) allowing for the creation, visualization and manipulation of ontologies in the OWL 1.0 format. As the new Protégé 4 has reimplemented its interface on top of the OWLAPI[26], which is already designed to support OWL 2[27], the ADL to OWL translator is also being adapted to the new API in order to get the benefits from the OWL 2 new features like Qualified Cardinality Restrictions (QCRs), see section 4.3.

2.3.2 SWRL Rules

Besides the biomedical interest in OWL, many health care processes such as computer aided decision making or disease diagnosis and treatment, are often best modelled using a declarative approach and rules, leading to a very active interest in rule-based systems (O'Connor et al., 2005). However, interoperability among the multitude of current rule-based systems is limited. The Semantic Web Rule Language (SWRL)[28] has emerged as a first step solution to increase rule-based systems interoperability from the Semantic Web perspective. It is based on a combination of OWL with the Rule Markup Language[29]. The combination of OWL and SWRL provides inference capabilities beyond the classification capabilities built-in the description logics implemented by OWL (Baader, Calvanese, McGuinness, Nardi, & P. Patel-Schneider, 2003). In the clinical environment,

[26] http://owlapi.sourceforge.net
[27] http://www.w3.org/TR/owl2-overview
[28] http://www.w3.org/Submission/SWRL
[29] http://www.ruleml.org

different kinds of rules can be expressed with this logic. For example, *standard-rules* allow for chaining ontologies properties as well as *mapping-rules* between ontologies contribute to data integration and navigability, as pointed out by Golbreich and Imai (2004). Currently SWRL provides a set of built-ins to deal with comparisons, math operations, strings and date/time among others. This modular approach to built-ins will allow further extensions in future releases and provides the flexibility for various implementations to select the modules to be supported with each version of the language. For instance, a SWRL extension to overcome complex scenarios that include mathematical relationships and formulas that exceed current SWRL capabilities is proposed by Sánchez-Macián, Pastor, Vergara, and López (2007).

Brief language introduction

SWRL semantics are based on OWL DL so it does not support direct reasoning about classes or properties. SWRL adopts the Open World Assumption[30]. A SWRL rule contains an antecedent part, which is referred to as the body, and a consequent part, which is referred to as the head. Both the body and head consist of positive conjunctions of *atoms*:

$$atom \;\wedge\; atom \;\ldots \quad \Longrightarrow \quad atom \;\wedge\; atom$$

While SWRL does not support negated *atoms* or negation as failure or disjunction, it does support classical negation. For example, *Programmer(?p)* is an atom where *Programmer* is an OWL named class, and *?p* is a variable representing an OWL individual. Informally, a SWRL rule may be read as meaning that if all the atoms in the antecedent are true, then the consequent must also be true. There are seven types of atoms, always of the form $\boldsymbol{p}(arg_1, arg_2, \ldots arg_n)$, i.e. a predicate $\boldsymbol{p}$ and its arguments:

- Class atoms
- Individual Property atoms

[30] The *Open World Assumption* is the assumption that the truth-value of a statement is independent of whether or not it is known by any single observer or agent to be true. It is the opposite of the *Closed World Assumption*, which holds that any statement that is not known to be true is false.

- Data Valued Property atoms
- Different Individuals atoms
- Same Individual atoms
- Built-in atoms
- Data Range atoms

In SWRL, the predicate symbols can include OWL classes, properties or data types. Arguments can be OWL individuals or data values, or variables referring to them. All variables in SWRL are treated as universally quantified, with their scope limited to a given rule.

SWRL built-ins are predicates that accept one or more data valued arguments. A number of core built-ins for mathematical and string operations are contained in the SWRL Built-in Proposal[31]. These built-ins are defined in the file *swrlb.owl*[32]. By convention, all core SWRL built-ins are preceded by the namespace qualifier swrlb. Examples of the use of SWRL[33] mathematical built-ins can be found in section 6.2.

SWRL rules and Clinical Guidelines

Clinical guidelines, also called medical guidelines, contain "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" (Field & Lohr, 1990). Examples of decision points and declarative knowledge contained in clinical guidelines that can be entirely or partially represented and shared by merging archetypes and rules are the following:

- **Antibiotic prescribing:** A complete example on respiratory tract infections is described by Lezcano, Sicilia, and Rodríguez-Solano (2011).
- **Risk assessment of pressure ulcers:** A key element in the Prevention and Treatment of the Pressure Ulcers guideline from NICE[34,35]. Figure 58 depicts some of the most important SWRL rules that were used in the CISEP project (see section 8.2) to evaluate pressure ulcers.

---

31 http://www.daml.org/rules/proposal/builtins.html
32 http://www.w3.org/2003/11/swrlb
33 For further details about SWRL visit http://www.w3.org/Submission/SWRL/
34 NICE - National Institute for Health and Clinical Excellence
35 http://guidance.nice.org.uk/CG29/

- **Alerts and risk of CVD:** Included in the key messages of the Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure[36,37].
- **Transverse Sinus Ligation:** The research by Lezcano, Sicilia and Serrano-Balazote (2008) was specifically oriented to aid intraoperative monitoring on *Transverse Sinus Ligation* by combining SWRL rules, like the one in Figure 42, with the OWL version of the *Intravascular Pressure* archetype.
- **Stages of COPD:** Defined in the Pocket Guide to COPD Diagnosis, Management, and Prevention[38,39]. SWRL rules to infer the stage of COPD according to such guideline are shown in Figure 59.

2.4 Clinical Terminologies

The consistent use of *clinical terminologies* and the development of good practice in archetype design and terminology binding to them play an important role in building structured EHRs and reaching semantic interoperability, as alleged by the European Commission (V. N. Stroetmann et al., 2009). The primary objective of clinical terminologies for interoperability is to enable the faithful exchange of meaning between machines and between machines and people. Therefore, a lot of research has to be done in order to integrate clinical terminologies with data models, ontologies, archetypes and the rest of knowledge artefacts involved in level 3 of SIOp. Some efforts have been already made to align the use of content in structured data models to one or more chosen terminologies (Markwell, Sato, & Cheetham, 2007; Smith & Ceusters, 2005).

In most cases the data and terminology models are developed independently by separate professional groups. While data modelling techniques are developed by IT professionals, terminologies are primarily dominated by clinicians and other clinical experts. Therefore, integration will be best achieved for terminologies by starting with areas where there is a high consensus on the need and the content. Translational medicine and *adverse drug reactions* are examples of such areas.

---

[36] CVD - Cardiovascular disease
[37] http://www.nhlbi.nih.gov/guidelines/hypertension/jncintro.htm
[38] COPD - Chronic Obstructive Pulmonary Disease
[39] http://www.goldcopd.com/guidelinesresources.asp

As the vocabulary surrounding terminologies and ontologies is sometimes confusing and different authors use the same terms differently, Rector (2007) provides a glossary that aids understanding by defining the terms in this context:

- *Controlled Vocabulary:* a list of specified items to be used for some purpose, usually in an information system to reduce ambiguity, misspellings, etc.
- *System of identifiers ("codes"):* Controlled vocabularies and many lexicons, ontologies and thesauri are usually accompanied by systems of identifiers for their units, e.g. typically, identifiers act as the primary unambiguous means of referring to the entities in the system for computational purposes with the text form being used for communication with users. Examples include the *Concept Unique Identifiers* (CUIs) from UMLS, the SNOMED identifiers, etc. In many contexts, identifiers are known as "codes."
- *Lexicon:* A list of linguistic units that may be attached to a controlled vocabulary or ontology, in a specific language or sublanguage, often including linguistic information such as synonyms, preferred terms, parts of speech, inflections and other grammatical material. Example: Term terms and lexical material in UMLS identified by Lexical Unique Identifiers LUIs).
- *Ontology (sensu information system):* a symbolic logical model of some part of the meanings of the notions used in a field, i.e. those things that are universally true or true by definition. The key relationship in an ontology is "subsumption" or "kind-of". Every instance of a subkind must be an instance of the kind, without exception. Typically, ontologies are implemented in logic languages such as Ontylog or OWL or frame systems such as Protégé-Frames. Examples: The GALEN Core Model, the stated form of SNOMED.
- *Classification:* an organisation of entities into classes for a specific purpose such as international reporting or remuneration. Examples ICD and Diagnosis Related Groups.
- *Thesaurus:* a system of terms organised for navigation with the primary relationship being "broader than"/"narrower than". The "broader than"/"Narrower than" relation is explicitly not limited to

subsumption/kind of relation. It is a general form of linguistic hyper/hyponymy aimed at assisting human navigation. However, it is explicitly not intended that it be used as the basis for logical interferences, e.g. in decision support. Examples MeSH, WordNet.

- *Knowledge Representation System / Background knowledge base:* the common knowledge to be assumed by the system, including both the ontology – what is universally true – and generalisations about what is typically true.
- *Terminology:* Any or all of the above in various combinations. Most health terminologies consist, at a minimum, of a controlled vocabulary and a system of identifiers. They may include extended lexicons, ontologies, thesauri or background knowledge base. This definition is deliberately broader and less specific than that in most of the standard references and intended to approximate common usage.
- *Coding system:* A terminology with attached identifiers or "codes".

2.4.1 Relevant Clinical Terminologies and Ontologies

SNOMED-CT

SNOMED-CT[40] is a comprehensive clinical terminology, originally created by the College of American Pathologists (CAP) and, as of April 2007, owned, maintained, and distributed by the International Health Terminology Standards Development Organization (IHTSDO), a non-for-profit association in Denmark. It provides the core general terminology for the EHR and contains more than 311,000 active concepts with unique meanings and formal logic-based definitions organized into hierarchies (Schulz, Suntisrivaraporn, Baader, & Boeker, 2009).

[40] http://www.ihtsdo.org/snomed-ct/

SNOMED-CT was selected for this research study because of the following reasons:

- As it is widely used in Europe and all over the world, this thesis could be significant for several researches in the EHR community. Spain has become the eleventh country to join the IHTSDO[41].
- SNOMED-CT has a large coverage of the clinical concepts required for the case studies that were carried out here.
- It is simple to use and query, in spite of its large size.
- At least part of the terminology has been classified by DL reasoners, making it more reliable when compared to completely unclassified terminologies. The reliability is based on the assumption that logical errors would have been resolved during classification. The semantic completeness of SNOMED-CT has been audited by Jiang and Chute (2009).

OpenGALEN

OpenGALEN[42] is a not-for-profit organisation providing another medical terminology. The GALEN programme of research into medical terminology began in 1991. In 1999 OpenGALEN was formed to provide an open source route both for disseminating the results of that programme and as a framework for its future development (Rector, J. E. Rogers, Zanstra, & Van Der Haring, 2003). The terminology is written in a formal language named GRAIL (GALEN Concept Representation Language). Currently available open source resources include a sophisticated ontology development environment and a large open source description logic-based ontology for the medical domain.

UMLS

The Unified Medical Language System (UMLS) is a compendium of three knowledge sources in the biomedical sciences, which are distributed with several tools that facilitate their use (Bodenreider, 2004; Lindberg, Humphreys, & McCray, 1993).

[41] http://www.ihtsdo.org/members/spain/
[42] http://www.opengalen.org/

The UMLS Metathesaurus is a large, multi-purpose, and multilingual vocabulary database that is organized by concepts. The current release comprises more than 1.5 millions biomedical terms from over 100 sources. Synonymous terms are clustered together to form a unique concept or cluster. Concepts are linked to other concepts by means of various types of relationships, resulting in a rich graph. The Semantic Network provides a consistent categorization of all concepts represented in the UMLS Metathesaurus as well as information about the set of basic Semantic Types, or categories, which may be assigned to those concepts. The Network contains 133 Semantic Types and 54 relationships.

The SPECIALIST Lexicon is a general English lexicon including many biomedical terms and the Lexical Tools are designed to address the high degree of variability in the natural language. Words often have several inflected forms which would properly be considered instances of the same word. The UMLS Knowledge Source Server (UMLSKS)[43] (Bangalore, Thorn, Tilley, & Peters, 2003), developed at the U.S. National Library of Medicine (NLM), is the set of machines, programs and APIs, written in Java, located and maintained by staff at the NLM that allows access to the UMLSKS services.

ICD

The International Classification of Diseases (**ICD**)[44] is a standard diagnostic classification for all general epidemiological, many health management purposes and clinical use. These include the analysis of the general health situation of population groups and monitoring of the incidence and prevalence of diseases and other health problems in relation to other variables such as the characteristics and circumstances of the individuals affected, reimbursement, resource allocation, quality and guidelines.

Recent efforts by S. W. Tu et al. (2010) include the implementation of the ICD-11 Content Model using OWL. This informal model is contains three layers: (i) the *Foundation layer* divided into the *Ontology layer*, which is intended to be aligned with a subset of SNOMED, and the *Category layer* that contains the description of each ICD category; (ii) the *Linearizations layer*, which is a generalization of the

---

43 http://umlsks.nlm.nih.gov/
44 http://www.who.int/classifications/icd/en/

traditional ICD classifications that provides the backwards compatibility (including their inclusions, exclusions, and residual categories) and supports new use cases. The ultimate goal is to develop Web-based software that allows wide participation in an expanded and enriched revision of the ICD.

In addition, there are other clinical terminologies and sources of knowledge that address specific areas of medicine. For example, the Logical Observation Identifier Names and Codes database (**LOINC**) is a naming structure for laboratory test (Forrey et al., 1996); **RxNorm**[45] is a standardized nomenclature for clinical drugs and drug delivery devices, produced by the National Library of Medicine (NLM); the Digital Imaging and Communication in Medicine (**DICOM**)[46] is a standard for medical imaging; and the Foundational Model of Anatomy (**FMA**)[47] is a reference ontology for the human anatomy. The BioPortal[48] allows to access and share ontologies that are actively used in biomedical communities, and RadLex[49] is a lexicon for uniform indexing and retrieval of radiology information resources.

2.4.2 Integrating archetypes and clinical terminologies

The `term binding` section in archetypes enable elements to be mapped to one or more terminology concepts, as shown in Figure 12, allowing to comply with terminology standards. This provides a means for a controlled method of data entry to enable reuse and lead to semantic interoperability. At present, such semantic mapping task is carried out manually to bind with external terminology systems such as ICD, SNOMED-CT and GALEN. Three main parts are required to create a term mapping statement in ADL. These are: (a) the internal fragment identifier, (b) the name of the terminology model such as LOINC, ICD or SNOMED, and (c) the terminology concept code.

Some of the terminology systems conceived by openEHR to be used with archetypes include: (a) SNOMED-CT, so that reliable inference and decision support based on EHR data can be made possible, (b) LOINC, so that traceability and sharing of laboratory data can be achieved, and (c) ICD and ICPC classifications, so

[45] http://www.nlm.nih.gov/research/umls/rxnorm/
[46] http://medical.nema.org/
[47] http://sig.biostr.washington.edu/projects/fm/
[48] http://bioportal.bioontology.org/
[49] http://www.radlex.org/

that reliable reimbursement, management, and public health studies can be made possible (Beale, Heard, Kalra, Lloyd, & Schloeffel, 2006).

This thesis combines archetypes with the OWL version of several SNOMED subsets including allergies, drugs and respiratory tract infections, among others. As recommended by Wroe (2006), the SNOMED concepts are considered as OWL classes that conform a subsumption hierarchy while nontaxonomic relations are represented by means of quantifier restrictions. A specific subset of SNOMED including allergies, drugs and respiratory tract infections is used. Clinical concepts that represent allergies are considered as OWL classes, as well as substances and drugs concepts. Then, the codes of those concepts will be considered as individuals or instances of the classes and used within SWRL rules.

3 Related Work

Many efforts have been made in the last decade to achieve interoperability of heterogeneous healthcare systems by means of terminologies, EHR models, clinical archetypes, templates, etc. Then ontologies and rules have also become an instrument to achieve such interoperability as the Semantic Web has progressively been adopted. More recently, the experience gained from those works have shown that dealing with SIOp requires the combination and seamless integration of several knowledge artefacts as none of them is able to completely fulfil the SIOp requirements when used in isolation. Thus, integration methods have started to appear in different areas of the clinical and biomedical domains. Some of them specifically designed to address a concrete necessity within current conditions, while other approaches have a wider scope and/or a future oriented application.

This chapter presents a range of research projects already completed or still being carried out to achieve semantic interoperability in EHR systems. Accordingly, references for three main combinations will be analysed. Firstly, those related to translating clinical archetypes to ontologies are presented. Then, some recently developed approaches to aid clinical guideline execution by combining ontologies and rules are described. Finally, the chapter concludes with the analysis of some projects addressing the integration of clinical terminologies and ontologies, as well as archetypes and terminologies bindings by means of ontologies. The discussion of each of the related works is completed with a comparative analysis with respect to this thesis' approach.

3.1 The POSEACLE approach

Translating ADL definitions to OWL can be done in two different ways. In the "*translating archetypes as classes*" method, ADL definitions can be considered as ontology classes that specialise the OWL representation of the RM. Therefore, the data about patients and clinical facts is represented as instances of those classes. Taking as example the *Heart_Rate-Pulse* archetype introduced in section 2.2.2, we can translate its components into a group of RM specialised classes (e.g. specialising the `OBSERVATION` class, the `ITEM_TREE` class, etc., see Figure 6).

A different approach or "*translating archetypes as instances*" takes archetypes as instances of an OWL representation of the AOM, leaving no room for patient data instances. When considering archetypes themselves as instance data, `ELEMENT`s, `ITEM_TREE`s, `CLUSTER`s, etc. are translated into instances of classes in the `Constraint_model` package shown in Figure 13. For example the `OBSERVATION` statement included in the *Heart_Rate-Pulse* archetype can be translated into an instance of the OWL representation of the `C_OBJECT` class (*complex object*) defined in the AOM.

The POSEACLE approach by Martínez-Costa, Menárguez-Tortosa, Fernández-Breis, and Maldonado (2009) has selected the former method, in which archetypes are translated into instances of some classes representing an archetype model. The main objectives of that project include facilitating semantic search at the archetype specification level, as well as other semantic tasks that improve EHR management. Their solution comprises the following steps: (i) creation of syntactic models representing ADL content; (ii) transforming syntactic models to semantic models conforming to CEN standard; and (iii) instantiation of OWL archetypes. In order to perform such steps, there is a need for an OWL ontology to represent clinical archetypes. The eAOM metamodel is defined for this purpose as an archetype representation that is common to all archetype standards.

In contrast with the POSEACLE approach, inference execution over recorded clinical data is the main final goal of the ADL to OWL translation method described in this thesis. Therefore, the former perspective (i.e. "*translating archetypes as classes*") is designed and explained in the following chapters, thus storing patient data as instances of RM specialised classes. The translation starting point is then an OWL representation of the RM instead of the AOM representation used in the POSEACLE approach.

3.2 The ARTEMIS project

The aim of the ARTEMIS project (Dogac et al., 2006) was to allow healthcare organizations to preserve their proprietary systems while exposing the functionality of their applications through Web Services. The Artemis Web service architecture proposed the use of ontologies to describe semantics but without requiring globally agreed ontologies; rather the reconciliation of healthcare institutes semantic differences was encouraged through a mediator component. Mediators were based on a P2P communication architecture to provide scalability and to facilitate the discovery of other mediators.

Clinical archetypes were translated to OWL, within the scope of the ARTEMIS project, for the purpose of achieving the interoperability of Web Service messages exchanged in the health care domain. Interoperability issues like creating semantic mappings between classes in different reference models (e.g. the EHRcom and the HL7 CDA) were addressed in the project. Thus, the OWL representation of archetypes was used to semantically annotate the Web Service messages and then provide the mapping between the OWL representations of archetypes through an OWL ontology mapping tool (OWLmt). Produced definitions were then used to automatically transform Web Service message instances when two healthcare institutes conforming to different archetypes wanted to exchange messages.

The translation principles explained in this thesis are consistent with the ones applied in the ARTEMIS project (Kilic, Bicer, & Dogac, 2005; Bicer, Kilic, Dogac, & Laleci, 2005). Going further, this thesis proposes workarounds to the translation issues that remain unsolved in those publications and, in addition, provides new translation mechanisms that take advantage of the improved potential of OWL 2. For example, regarding the translation of the data constraints contained in archetypes, a complete description containing the limitations of OWL 1 datatyping can be found in a study by J. Z. Pan and Horrocks (2005). They proposed an extension to OWL DL, called OWLEu, which integrates a large family of decidable

Description Logics with unary datatype groups, so as to support user defined datatypes. However, the emerging OWL 2 adopted a different approach that is used in section 4.2 to capture the quantitative constraints of archetypes. Another significant difference with the ARTEMIS approach to archetypes translation is that the methods in Chapter 4 explain not only how to represent the ADL constraints and semantics in OWL but also how to accomplish such task without human intervention. Thus, they were conceived as completely automatic processes that support the implementation of the automatic ADL to OWL translator described in Chapter 5.

3.3 Clinical guidelines integration

According to the National Institute for Health and Clinical Excellence (NICE)[50], clinical guidelines aim to improve the quality of healthcare. They can change the process of healthcare and improve people's chances of getting as well as possible. Clinical guidelines main objectives are to:

- provide recommendations for the treatment and care of people by health professionals
- be used to develop standards to assess the clinical practice of individual health professionals
- be used in the education and training of health professionals
- help patients to make informed decisions
- improve communication between patient and health professional

Given the each day more important interoperability feature between healthcare systems and the increasing use of Semantic Web technologies within services integration, there have been a few approaches addressing the incorporation of clinical guidelines in healthcare systems to support the decision making process by means of these technologies. For example, the approach by Argüello and Des (2008) explains how to facilitate the use of a clinical guideline for the diagnosis and clinical management of Diabetic Retinopathy by means of web services. The approach

[50] http://www.nice.org.uk/

outlines the use of the OWL's XML syntax to obtain web services that provide reasoning and easily deal with fact and rules, which are defined in SWRL.

Their proposed implementation considers three Web services: (i) the *Patient Identification service*, (ii) the GL clinical information service that finds a relevant Clinical Practice Guideline and gathers the required clinical information about the patient, and (iii) the *GL recommendation service* that evaluates the patient condition and makes recommendations about the clinical management based on the evidence available. The inputs and outputs of these web services are based on a group of ontologies that describe the require concepts: (i) the SWRC ontology[51], (ii) the *Organization Extension ontology* that extends the *SWRC ontology* and reuses a semantic type from UMLS, (iii) the Document Extension ontology which is an extension of the SWRC ontology to include Clinical Practice Guidelines, and (iv) the Data Set ontology which is introduced to encode the OWL domain ontology fragments and the SWRL fragments with an XML presentation syntax. Authors argue that having inference mechanisms and descriptive knowledge combined under the same syntactic structure provides means for the interoperability of rule systems.

The more recent work by C. Chen, K. Chen, Hsu, and Li (2010) has also been carried out to integrate clinical guidelines by means of ontologies and rules. The research describes an application based on Protégé and Java technologies aimed at translating the visual representation of clinical guidelines rules to a representation in XML, which in turn is transformed to Jess rules for execution. Although their results show high levels of effectiveness when tested against historical data, the rules derived from the guidelines and the used data instances are not expressed in a language that is independent from their particular implementation.

In parallel to these Semantic Web based approaches, traditional designs have been also appearing. The system presented by Rossille, Laurent, and Burgun (2005) is meant to be a data warehouse in oncology, storing valuable information for treating, for instance, patients with rare tumours, or not reacting normally to a treatment. It is a multi-modal decision-support system as it is based on both rule-

[51] http://ontoware.org/swrc/

based reasoning (with GLIF3[52] guidelines) and case-based reasoning (with individual cases).

In order for such system to be suitable to analyse any tumour, the case has an object-oriented architecture composed of classes like *Patient*, *Alarms*, *FamilyHistory*, *Episode*, *CharacteristicsCancer*, *CharacteristicsMetastasis*, *Treatments (CT, RT, HT, Surgery)* and *Exams*, to which cancer-dependent classes are associated or inherited (such as *BreastCancer* characterizing the primary breast tumour, or *BreastFactors* characterizing the episode-independent breast cancer dependent factors).

As can be seen, all the approaches introduced in this section had to select a mechanism to represent guidelines steps and structures as well as a knowledge representation mechanism to represent patients' data. GLIF is sometimes used to model clinical guidelines while mappings to clinical terminologies like UMLS and ICD is used in other cases to reduce discrepancies with regard to clinical concepts meanings. However, even when some importance has been given to knowledge normalization, standards are not equilibrated in none of above mentioned works. As a result, either patients' data or clinical guidelines are always implemented through a specific or ad-hoc approach the makes the overall implementation dependant from a particular context, underlying systems and locally agreed semantics. This entails sharing and reusing difficulties that preclude reaching level 3 of SIOp (V. N. Stroetmann et al., 2009). In contrast, the research described in this thesis is addressed to avoid these inconveniences by completely relying on the integration of widely accepted models and standards.

[52] The *Guideline Interchange Format (GLIF)* is a computer-interpretable language for modelling and executing clinical practice guidelines. GLIF supports sharing of computer-interpretable clinical guidelines across different medical institutions and system platforms. GLIF has a formal representation. It defines an ontology for representing guidelines, as well as a medical ontology for representing medical data and concepts. Tools are under development to support guideline authoring and execution. http://mis.hevra.haifa.ac.il/~morpeleg/Intermed/

3.4 Binding terminologies to clinical archetypes

Previous work on terminology mappings includes two separate tracks. On one hand is the problem of finding suitable terminology codes to map to the archetype terms, while on the other hand is the problem of testing the logical correctness of these mapped codes. A relevant research addressing the first issue is the Model Standardisation using Terminology (MoST) (Qamar, 2008), which is largely involved with finding the semantically correct SNOMED codes to bind to the archetype fragments.

Significant works in the other line of research include a general methodology for defining a code binding interface in OWL (Rector, Qamar, & Marley, 2009). Such Code Binding Interface constrains and specifies how coding systems are to be used in EHR data structures. The approach presents the development of an ontology that acts as a meta-model of meaning, defining the codes that can be logically included in place of the archetype fragments, to retain the semantic and logical correctness of the original data model. The intention is to classify the ontology to indicate any inconsistencies in the integrated model. It should be noted that this task is essentially syntactic as it is concerned with the reliable processing of data structures and not with whether the information conveyed is accurate and correct. The authors argue that many controversies around coding systems and EHR standards arise from the lack of a clear distinction between validity and accuracy. The approach needs to be further evaluated in order to become widely accepted as a key step towards semantic interoperability.

In that second direction, Chapter 7 illustrates several applications and advantages that arises from a similar binding when applied to the case of openEHR, OWL and SNOMED. Such benefits are supported by the ADL to OWL translation approach that is explained in this thesis before arriving to Chapter 7. Also some issues related to the connections between Information, Terminology and Inference Models have been studied by Rector, Johnson, S. Tu, Wroe, and J. Rogers (2001). Taking the OWL version of archetypes as metadata allows for ontologies to be used to index these clinical statements and provide better tools of retrieval. There have already been several research efforts that address the indexing of information by means of ontologies (Tzitzikas, 2002), and according to Kalra (2008), archetypes indexing and archetypes repository services fall into the areas needing research inside the semantic interoperability domain. With the increase in the amount of definitions, archetypes management will become a key matter of concern.

4 Overall approach to the translation

This chapter describes the approach resulting from this research to translate the representation of clinical archetypes in ADL to an ontology language like OWL. Section 4.1 exposes general aspects of the automatic translation from ADL to OWL. Then section 4.2 details the mappings of quantitative constraints to the ontological version of the archetype. Section 4.3 gives a recommendation on how to translate the occurrences constraints. Then, section 4.4 describes the importance of annotation properties as a means for a reusing technique and section 4.5 shows a method for constraining an archetype term to a small and finite set of values. Finally, section 4.6 provides a list of the basic translations and mapping rules applied in the translation of leaf data and ADL constraints keywords.

4.1 Overview of the ADL to OWL translation

ADL files express clinical archetype definitions by means of constraint definitions (*cADL*) and data definitions (*dADL*) as explained in section 2.2.2. Therefore, translating ADL files to OWL implies rendering each and every *cADL* and *dADL* definitions using the OWL syntax. Main difficulties underlying this process come from the fact that there are different ways to represent the same information in OWL. This research requires a proper mechanism to be selected in order to support further tasks like adding SWRL rules and launching inference, setting bindings to terminologies, validating archetypes, etc. Thus, it should be noted that describing constraints as human-readable comments should be avoided as far as possible given that they cannot be used by semantic reasoners. For example, numerical range constraints, which are very common in archetype definitions, should not be translated as annotation properties but as user-defined datatypes (see section 4.2).

The two-level modelling paradigm and the knowledge representation mechanism followed by archetypes are analogous to some Object-Oriented Programming (OOP) patterns. As well as a *Student* class is frequently defined in an OOP context as a specialisation of a more general *Person* class, archetypes define clinical concepts by specialising more general `ENTRY` types and other classes from the RM. Therefore, the applied ADL to OWL translation implements such *inheritance* relation between the AM and the RM by means of the fundamental taxonomic constructor *rdfs:subClassOf*. It relates a more specific class to a more general class. If an OWL class **A** is a subclass of **B**, then every instance of **A** is also an instance of **B**. The *rdfs:subClassOf* relation is transitive and reflexive (i.e. every class is its own subclass, for example every person is a person). As with OOP inheritance, the OWL version of an archetype inherits all attributes and properties from its parent RM class and is able to overwrite them by adding new restrictions. For example, narrowing the allowed *Blood Pressure* numerical values to the range of integers between 0 and 250 follows similar purposes to those of the bundling data mechanism provided by OOP *encapsulation*.

Given the archetypes capability of being defined as specialisations of more general archetypes, the translation mechanism encourages the compatibility of SWRL across hierarchies. Thus, an SWRL rule originally designed according to the OWL version of a parent archetype will be suitable for all its descendants. This "*Inheritable compatibility*" as well as other benefits of the OWL + SWRL integration proposed in this thesis are all listed in section 6.1. The principles of this *"translating archetype as classes"* approach will be detailed throughout the current chapter. Existing alternative approaches are commented in the Related Work Chapter.

During translation we deal with two different information hierarchies. One of them is defined by the archetype level in an ADL file as shown in section 2.2.2. This one is composed of RM containers and classes from the constraint model package (Figure 13). This containers-tree has a variable structure because it depends on the objectives of specific clinical situations. The other one is the hierarchal tree conforming the RM, which has the same structure across different archetypes. The approach proposed in this thesis considers that each level of the archetype tree defines a subclass of an entity belonging to the RM tree. So, to map the archetype tree to OWL we need to specialise the corresponding RM classes while preserving the archetype tree interconnections between new classes.

It should be noted that both trees are mixed in a way that remains compatible with the RM original structure. The point of departure is the mapping developed by Román, Roa, Reina-Tosina, and Madinabeitia (2006). Archetype definitions start with an `ENTRY` subtype like `EVALUATION`, `INSTRUCTION`, `ACTION` or `OBSERVATION`. Essentially, an archetype constrains the instances of such categories, so a main translation principle is having those `ENTRY` categories as classes and each archetype definition becoming a subclass depending on the subtype.

This section takes as example the translation process of the *Heart rate and rhythm* archetype. It is an `OBSERVATION` to record the measured characteristics related to the rate and rhythm of the heart, including a simple statement of presence of heart rate. These are not recorded by direct observation of the heart itself but inferred from alternative sources including the direct auscultation of the heart or an electrocardiograph reflecting the electrical activity of the heart.

In general, the `OBSERVATION` type is used to record the observation of any phenomenon or state of interest related to the patient. It only records information relating to the situation of the patient, not what was actually done while treating him/her. Among other things, observations also include pathology analysis results as well as the family history and social circumstances of the patient as told to the clinician. *Heart rate and rhythm* (or its specialisation, *Pulse*) are commonly recorded as one component of *Vital signs*, which includes: *Blood Pressure*, *Respirations*, *Temperature* and *Oximetry*. There are additional specific `OBSERVATION` archetypes for each of these concepts.

In addition to the attributes inherited from `ENTRY` and `CARE ENTRY`, the `OBSERVATION` type has only two attributes, `data:HISTORY` and `state:HISTORY`.

The translation process and principles illustrated through an `OBSERVATION` can be then applied to any sort of archetypes.

```
definition
  OBSERVATION[at0000] matches {       -- Heart rate and rhythm
    data matches {
      HISTORY[at0002] matches {       -- history
```

Figure 16. *Heart rate and rhythm* archetype (ADL root).

```
<owl:Class rdf:ID="Heart_rate_and_rhythm">
  <NodeID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >at0000</NodeID>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:resource="#HISTORY"/>
      <owl:onProperty rdf:resource="http://.../EHR/EHR_RM.owl#data"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://.../EHR/EHR_RM.owl#OBSERVATION"/>
</owl:Class>
```

Figure 17. *Heart rate and rhythm* archetype (OWL XML/RDF syntax).

```
CLASS Airway_assessment_anaesthesiology
  ANNOTATIONS: NodeID "at0000"
  SUBCLASSOF: OBSERVATION THAT data ONLY HISTORY
```

Figure 18. *Heart rate and rhythm* archetype (OWL Manchester syntax).

The code fragments provided in Figure 16 to Figure 18 illustrate the mapping from the archetype root definition to the *Heart rate and rhythm* OWL class. The *rdfs:subClassOf* property is being used to inherit all the `OBSERVATION` features. The name of the new OWL class is retrieved from the ADL `ontology` section using the ADL node identifier, in the case of the example, `[at0000]`. This code is attached to the class by an *annotation property* named *NodeID*. Annotation properties are pieces of metadata that can place annotations on classes. Other groups of properties can only relate data values, individuals and ontologies[53].

As well as translating each ADL node to an OWL class we need to preserve the connections established by ADL relations like data. By translating these ADL relations to OWL individual-valued properties and restricting them by means of an *owl:allValuesFrom* property, we guarantee that defined classes exactly map the configuration of the structure in the archetype specification. Also known as *universal* restriction, the *owl:allValuesFrom* is a built-in OWL property that constrains the relationships along a given property to individuals that are members of a specific class (Horridge, 2009).

53 http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/#Annotation_Properties

Figure 19 illustrates how the OWL root class have a restriction on its *data* property that forces class instances to be related only to *History* instances, which is the next downwards archetype definition. At the same time, each OWL class is created as a specialisation of its RM category. Because archetypes have a tree like structure the above steps are repeated for each level until the whole hierarchy is mapped.

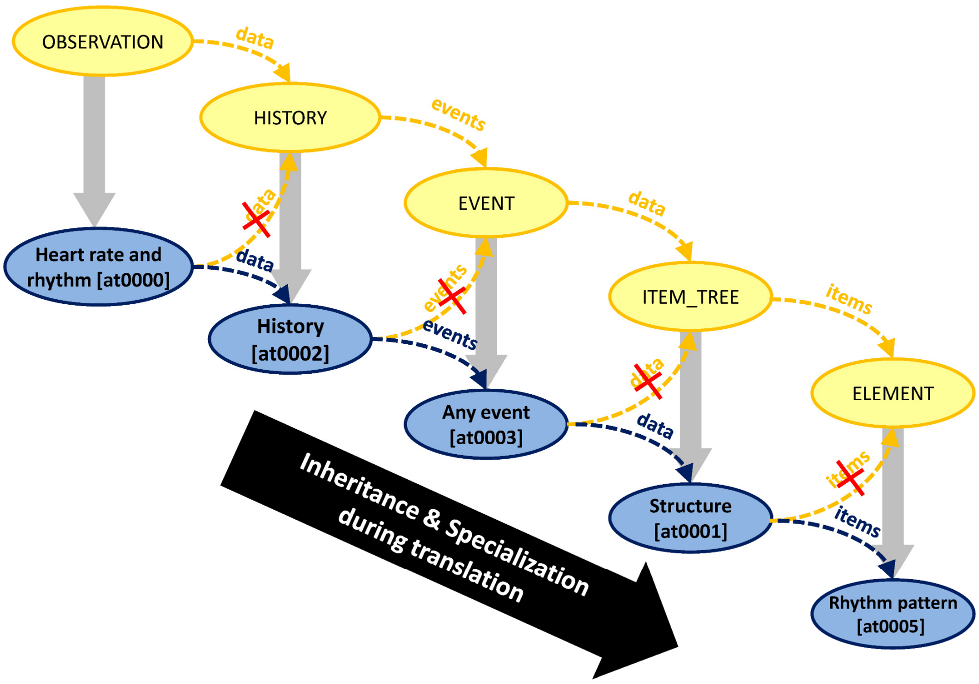

Figure 19. Restriction of inherited properties to guarantee OWL and ADL structure compatibility.

Considering the ADL tree as a graph, the traverse method applied by the translation process resembles the *Depth First Search* pattern or *DFS*. It extends the current path as far as possible before backtracking to the last choice point and trying the next alternative path. A deeper explanation can be found in Cormen, Leiserson, Rivest, and Stein (1990).

In this particular example, the yellow ovals represent the RM classes being restricted by the *Heart rate and rhythm* archetype. The blue ovals represent the OWL subclasses implementing those restrictions. The Xed out yellow dotted lines and the blue dotted lines represent the addition of new *owl:allValuesFrom* restrictions that override the inherited ones and confine the bindings to the subclasses defined for this archetype. Thus, the archetype tree-like structure is being simulated on top of the RM.

At the ADL's bottommost level we find several types of data-valued constraints (e.g. `C_DATE`, `C_BOOLEAN`, etc.) that are translated to their counterparts in the RM ontology whose names start with DV (e.g. `DV_BOOLEAN`, `DV_DATE`). The following sections explain the particular cases where translation is more complicated.

A mapping reference including the formal rules for these translation techniques is listed in section 4.6.

4.2 User-defined datatypes to represent restrictions on quantified values

Archetypes allow a wide variety of constraints to be applied on the primitive types located at the ADL's bottommost level. For example, in the case of indicators like rates, temperatures, indexes and pressures, constraints are commonly needed to represent limits in measurement. The OWL 1.0 language presents serious disadvantages to cover this kind of restriction. It allows controlling the cardinality of relations using the *owl:cardinality* built-in[54] and also it may guarantee the link with a certain value using the *owl:hasValue* built-in[55]. By combining both restrictions we can constrain a primitive type to a small and discrete set. However, it is not enough to deal with a continuous range defined by a minimum and a maximum value. A complete description containing the limitations of OWL 1.0 datatyping can be found in Pan and Horrocks (2005).

---

[54] A restriction containing an *owl:cardinality* constraint describes a class of all individuals that have exactly ***N*** semantically distinct values (individuals or data values) for the property concerned, where ***N*** is the value of the cardinality constraint. http://www.w3.org/TR/owl-ref/#cardinality-def

[55] A restriction containing an *owl:hasValue* constraint describes a class of all individuals for which the property concerned has at least one value semantically equal to a given *individual* or a *data value* (it may have other values as well). http://www.w3.org/TR/owl-ref/#hasValue-def

To overcome this issue, there are a number of approaches:

- Pan and Horrocks (2005) and Pan (2004) describe an extension to OWL DL, called *OWLEu*, that integrates a large family of decidable *Description Logics* with unary datatype groups, so as to support user defined datatypes.
- There have been approaches from the language designers that tend towards reusing the mechanism for user defined datatypes in the XML Schema specifications[56]. Since the XML Schema blocks are not in RDF format, there is no consensus on what URI should be used to identify the defined datatypes. Therefore they cannot easily be integrated in the ontology. Options to this problem have been drafted by Carroll and Pan (2006).
- Another approach is provided by Knublauch (2005), explaining a Protégé-OWL implementation of user defined datatypes. It involves the use of a small extension ontology *xsp.owl* that defines RDF properties to represent XML schema facets. Once imported in our ontology file, user-defined datatypes can be embedded into the same file in contrast with the two files required by the above proposal.
- OWL 2 has adopted a very similar solution to the one proposed by Knublauch, in order to deal with user-defined datatypes. The normative constraining facets[57] for the datatype are *xsd:minInclusive*, *xsd:maxInclusive*, *xsd:minExclusive*, and *xsd:maxExclusive*.

Currently the translation implementation applies the penultimate solution given its simplicity and consistency for the automatic translation process and because it is fully supported by the Protégé-OWL API. However, the implementation is being migrated in order to apply the last solution.

```
ELEMENT[at0.11] occurrences matches {0..1} matches {   -- Missed beats per minute
  value matches {
    DV_COUNT matches {
      magnitude matches {|>=0|}
    }
  }
}
```

Figure 20. A DV_COUNT constraint from the *Pulse* archetype (ADL).

---

56 http://www.w3.org/TR/xmlschema-2/

57 http://www.w3.org/TR/owl2-syntax/#Real_Numbers.2C_Decimal_Numbers.2C_and_Integers

```
<rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  <xsp:minInclusive rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >0</xsp:minInclusive>
  <xsp:base rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</rdfs:Datatype>
```

Figure 21. A `DV_COUNT` constraint from the *Pulse* archetype (OWL XML/RDF syntax).

```
DATAPROPERTY: magnitude
       RANGE: integer [>= 0]
```

Figure 22. A `DV_COUNT` constraint from the *Pulse* archetype (OWL Manchester syntax).

An example from the *Pulse* archetype can be appreciated in Figure 20. The *Pulse* contains the `ELEMENT[at0.11]` that outlines the *Missed beats per minute* as an integer greater than or equal to 0. Its corresponding OWL code according to Knublauch recommendation is given in Figure 21 and Figure 22.

4.3 Representing *occurrences* restrictions in OWL

The ADL syntax includes the `occurrences{...}` statement that is used only with cADL object nodes (not attribute nodes) to restrict the times that a particular piece of information is recorded as part of a more general container. For example, `occurrences ∈ {1..1}` indicates that the constrained object is mandatory while `occurrences ∈ {1..0}` indicates that it is optional.

An `occurrences{...}` statement differs from a `cardinality{...}` one in the sense that it affects the contained structures themselves instead of the whole collection count. Both occurrences and cardinality are equivalent restrictions only in the case where the container includes a single type of structure. For the rest, the `occurrences` restriction is stronger than `cardinality`. There is a rule[58] defined by Beale and Heard (2008b) for the purpose of validating the relation between both restrictions: *the interval represented by <the sum of all occurrences minimum values> .. <the sum of all occurrences maximum values> must be inside the interval of the cardinality.*

---

[58] The ADL Validity Rules describe formal and checkable semantics of archetypes. It is recommended that parsing tools use the identifiers published here in their error messages, as an aid to archetype designers.

```
ITEM_TREE[at0001] matches {   -- structure
  items cardinality matches {0..*; unordered} matches {
    ELEMENT[at1005] occurrences matches {0..1} matches {        -- Heart rate present
      value matches {...}
    }
    ELEMENT[at0004] occurrences matches {0..1} matches {        -- Rate
      value matches {...}
    }
    ELEMENT[at0005] occurrences matches {0..1} matches {        -- Rhythm pattern
      value matches {...}
    }
    ELEMENT[at0009] occurrences matches {0..1} matches {        -- Comment
      value matches {...}
    }
  }
}
```

Figure 23. `occurrences` restrictions in the *Heart rate and rhythm* archetype (ADL).

```
<owl:Restriction>
  <owl:onProperty
  rdf:resource="http://.../Data_Structures_RM.owl#items"/>
  <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</owl:maxCardinality>
</owl:Restriction>
```

Figure 24. A constraint on the items cardinality (OWL XML/RDF syntax).

Despite the fact that OWL 1.0 lacks of a direct mechanism to simulate the ADL `occurrences` statement, there is a workaround based on logical class operations that fulfils it. The following procedure takes as example the `ITEM_TREE` shown in Figure 23.

i. Each `occurrences` restriction is first considered as if it were the only one inside the container so it can be replaced by a `cardinality` restriction on the container. For example, the `occurrences` restriction on the *Rhythm pattern* `ELEMENT` is substituted by a `cardinality` restriction on the `items` property of the `ITEM_TREE` and expressed in OWL as appears in Figure 24.
ii. Then the `cardinality` restriction is merged with a *universal* restriction to guarantee the correct `ITEM_TREE<->ELEMENT` link as explained in section 4.1.
iii. Repeat the above steps for every other structure belonging to the container (i.e. *Heart_rate*, *Rate* and *Comment* `ELEMENTs`).

iv. Finally, all universal-cardinality restriction pairs must be joined using an *intersection* closure to create an *anonymous* class[59] that will become a superclass of the container class.

It should be noted that the container cardinality that was originally defined in the ADL code is no longer significant after applying the above mentioned steps.

Nevertheless, the selected solution to capture the semantics of the ADL `occurrences` restrictions is based on Qualified Cardinality Restrictions (QCRs)[60], provided by the latest version of OWL. QCRs are suitable to represent `occurrences` because the term qualified indicates that they apply only to a specific type of value rather than to the property overall. Taking as example the `ITEM_TREE` shown in Figure 23, those `occurrences` restrictions can be translated using QCRs and represented in the OWL Manchester Syntax as in Figure 25.

Alternative workarounds based on *subproperties* were discarded because they do not enforce the property to be used only through one of its subproperties, thus allowing for the archetype tree-like structure to be invalid. This kind of workaround was discussed by Rector and Schreiber (2005).

```
CLASS Structure
  SUBCLASSOF: items ONLY (Heart_rate_present OR Rate OR Rhythm_pattern OR Comment)
  SUBCLASSOF: items MAX 1 Heart_rate_present
  SUBCLASSOF: items MAX 1 Rate
  SUBCLASSOF: items MAX 1 Rhythm_pattern
  SUBCLASSOF: items MAX 1 Comment
```

Figure 25. ADL occurrences statements expressed in OWL Manchester Syntax.

[59] *Anonymous* classes or *unnamed* classes are described by a restriction. Hence the anonymous class contains all of the individuals that satisfy that restriction, (Horridge, 2009).

[60] While OWL 1 allows for restrictions on the number of instances of a property, e.g., for defining persons that have at least three children, it does not provide a means to restrain the class or data range of the instances to be counted (*qualified cardinality restrictions*), e.g., for specifying the class of persons that have at least three children who are girls. In OWL 2, both qualified and unqualified cardinality restrictions are possible. http://www.w3.org/TR/owl2-new-features/

4.4 A reuse technique based on annotation properties

As every other information model, archetypes need to exploit the advantages of reusing previously defined CLUSTERs, ITEM_TREEs, ELEMENTs, etc. in order to make the entire definition more efficient and less redundant. The use_node ADL keyword has been created for this purpose. It works by referencing an ADL node from the location where it should be repeated. A reliable path that identifies the node in the ADL text (concatenating the [atXXXX] IDs of its containers) serves as internal reference. As explained in section 4.1, those term IDs are mapped in OWL as the annotation property *NodeID* that links the OWL classes with their ADL ID. It is the annotations peculiarity of taking classes as operands instead of instances who led this thesis approach to use them in the implementation of the reusing technique.

```
CLUSTER[at0010] occurrences matches {0..*} matches {   -- Localised palpation
  items cardinality matches {1..*; unordered} matches {
    ELEMENT[at0011] matches { -- Body site
      value matches {
        DV_TEXT matches {*}
    }
}
________________________________________________________________________________

CLUSTER[at0019] occurrences matches {0..*} matches {   -- Tenderness
  items cardinality matches {1..*; unordered} matches {
    ELEMENT[at0002] occurrences matches {0..1} matches {...}    -- Depth of palpation
    ELEMENT[at0020] occurrences matches {0..1} matches {...}    -- Degree
    ELEMENT[at0005] occurrences matches {0..1} matches {...}    -- Category
    use_node ELEMENT /items[at0080]/items[at0010]/items[at0011] -- reference to Body_site
  }
}
```

Figure 26. use_node keyword in the *Palpation* archetype.

A fragment from the *Palpation* archetype that includes an example of a use_node statement and the referenced ELEMENT is shown in Figure 26. The purpose of the archetype is to record data found on examination through palpation. Within the archetype specification, the *Body_site* ELEMENT must be recorded twice (i.e. in the *Localised_palpation* CLUSTER and in the *Tenderness* CLUSTER) and therefore the use_node keyword is included to allow a single definition and a double use. When parsing an ADL file, use_node keywords are interpreted as instances of the ArchetypeInternalRef AOM class. This class supports the finding process of the archetype node that is going to be reused. Figure 27 contains the translation into OWL of the *Body_site* ELEMENT (i.e., the referenced object in this case). In order to map the *use_node* statement to OWL, the container structure (i.e. the *Tenderness* CLUSTER) is linked to the referenced node. The OWL fragments in Figure 27 and Figure 28 guarantees such linkage by means of the *owl:allValuesFrom* restriction.

```
<owl:Class rdf:about="#Body_site">
  <NodeID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >at0011</NodeID>
  <rdfs:subClassOf rdf:resource="http://.../Data_Structures_RM.owl#ELEMENT"/>
  ...
</owl:Class>
___________________________________________________________________________

<owl:Class rdf:ID="Tenderness">
  ...
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://.../Data_Structures_RM.owl#items"/>
        <owl:allValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#Depth_of_palpation"/>
              <owl:Class rdf:about="#Degree"/>
              <owl:Class rdf:about="#Category"/>
              <owl:Class rdf:about="#Body_site"/>
            </owl:unionOf>
     ...
  ...
</owl:Class>
```

Figure 27. Reusing the *Body_site* class in the *Tenderness* CLUSTER (OWL XML/RDF syntax).

```
CLASS Tenderness
  SUBCLASSOF: items ONLY (Depth_of_palpation OR Degree OR Category OR Body_site)
  SUBCLASSOF: items MAX 1 Depth_of_palpation
  SUBCLASSOF: items MAX 1 Degree
  SUBCLASSOF: items MAX 1 Category
  SUBCLASSOF: items MAX 1 Body_site
```

Figure 28. Reusing the *Body_site* class in the *Tenderness* CLUSTER (OWL Manchester syntax).

4.5 Constraining possible values to a finite and small set

Pieces of data that must be recorded as text fragments are usually instances of DV_TEXT. This RM class is used to contain any amount of legal characters arranged as words, sentences, etc. However, if a controlled vocabulary or terminology is required, then the DV_CODED_TEXT class allows the definition of value sets in groups using codes and rubrics. Codes themselves are contained within the defining code attribute of the class.

Other situations require the recording of symbolic values when exact values are not of interest or they are unknown. The main purpose is usually to classify patients into fuzzy intervals for which different decisions might be made. Take for example the categorization of *Tenderness* as being `"Superficial tenderness"`, `"Deep tenderness"`, `"Rebound tenderness"`, `"Tenderness"` or `"Rigidity"`. This is what `DV_ORDINAL` class is designed for.

```
ELEMENT[at0005] occurrences matches {0..1} matches {   -- Category
  value matches {
    DV_CODED_TEXT matches {
      defining_code matches {
             [local::
             at0007, -- Superficial tenderness
             at0008, -- Deep tenderness
             at0009, -- Rebound tenderness
             at0019, -- Tenderness
             at0025] -- Rigidity
      }
    }
  }
}
```

Figure 29. The Category `ELEMENT` in *Palpation* archetype.

In both cases the translation process must ensure that instances from the resulting OWL class can only take values among the listed finite set. The procedure is here elucidated working from the *coded text* in Figure 29. We must specialise the OWL version of the `ELEMENT` class to obtain a *Category_ELEMENT* class that only accepts instances from a given specialisation of `DV_CODED_TEXT`, which name is automatically generated as *Category_DV_CODED_TEXT*. Such class will in turn be linked to a specialisation of the `CODE_PHRASE` class that only accepts the codes: `at0007`, `at0008`, `at0009`, `at0019` or `at0025`. To represent all possible values, an *anonymous* class is created using the union closure and it is established to be a superclass of the *Tenderness_Category_CODE_PHRASE* class. This way every instance fulfilling at least one of the restrictions specified in the union may be considered as a member of the *Tenderness_Category_CODE_PHRASE* class. Then the code set is traversed in order to add every possible value to the union closure. Finally, as rubrics are human readable information (i.e. `"Deep tenderness"`, `"Rigidity"`, etc.) they are connected to the class through an annotation comment. Figure 30 and Figure 31 show the resulting OWL definition.

```
<owl:Class rdf:about="#Tenderness_Category...">
  <rdfs:subClassOf rdf:resource="http://...#CODE_PHRASE"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://...#code_string"/>
```

```
            <owl:hasValue rdf:datatype="http://.../XMLSchema#string"
            >at0007</owl:hasValue>
          </owl:Restriction>
          <owl:Restriction>
            <owl:hasValue rdf:datatype="http://.../XMLSchema#string"
            >at0008</owl:hasValue>
            <owl:onProperty rdf:resource="http://...#code_string"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:hasValue rdf:datatype="http://.../XMLSchema#string"
            >at0009</owl:hasValue>
            <owl:onProperty rdf:resource="http://...#code_string"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:hasValue rdf:datatype="http://.../XMLSchema#string"
            >at0019</owl:hasValue>
            <owl:onProperty rdf:resource="http://...#code_string"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:hasValue rdf:datatype="http://.../XMLSchema#string"
            >at0025</owl:hasValue>
            <owl:onProperty rdf:resource="http://...#code_string"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:Class>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://.../XMLSchema#string"
    >at0007 -&gt; Superficial tenderness, at0008 -&gt; Deep tenderness,
     at0009 -&gt; Rebound tenderness, at0019 -&gt; Tenderness, at0025 -&gt; Rigidity
    </rdfs:comment>
  </owl:Class>
```

Figure 30. Specialisation of `CODE_PHRASE` to represent *Tenderness* categories (OWL XML/RDF syntax).

There is another workaround for this translation that is based on the *owl:allValuesFrom*. At first a new Enumerated Datatype[61] is created, including all accepted codes. Then the `code_string` property is restricted using an *owl:allValuesFrom* so all related data values are within the data range that has been created. This second procedure does not require a union operation. Although both approaches have the same effect, the first one has been selected for implementation issues.

```
CLASS Tenderness_Category_CODE_PHRASE
  ANNOTATIONS: rdfs:comment "at0007 -> Superficial tenderness, at0008 -> Deep tenderness,
                             at0009 -> Rebound tenderness, at0019 -> Tenderness,
                             at0025 -> Rigidity"
  SUBCLASSOF: CODE_PHRASE
  SUBCLASSOF: code_string VALUE "at0007" OR code_string VALUE "at0008" OR
              code_string VALUE "at0009" OR code_string VALUE "at0019" OR
              code_string VALUE "at0025"
```

Figure 31. Specialisation of `CODE_PHRASE` to represent *Tenderness* categories (OWL Manchester syntax).

[61] http://www.w3.org/TR/owl-ref/#EnumeratedDatatype

4.6 Translation Reference

This reference includes a list of the basic translations and mapping rules applied in the translation of leaf data and ADL constraints keywords. As pointed out below, the translation of many common data types from their ADL representation (dADL) to an OWL compatible representation is straightforward. The *W3C XML Schema*[62] and *RDFS Literals*[63,64] have been chosen for this purpose. The hierarchy structure in Figure 32 illustrates the XML Schema built-in datatypes.

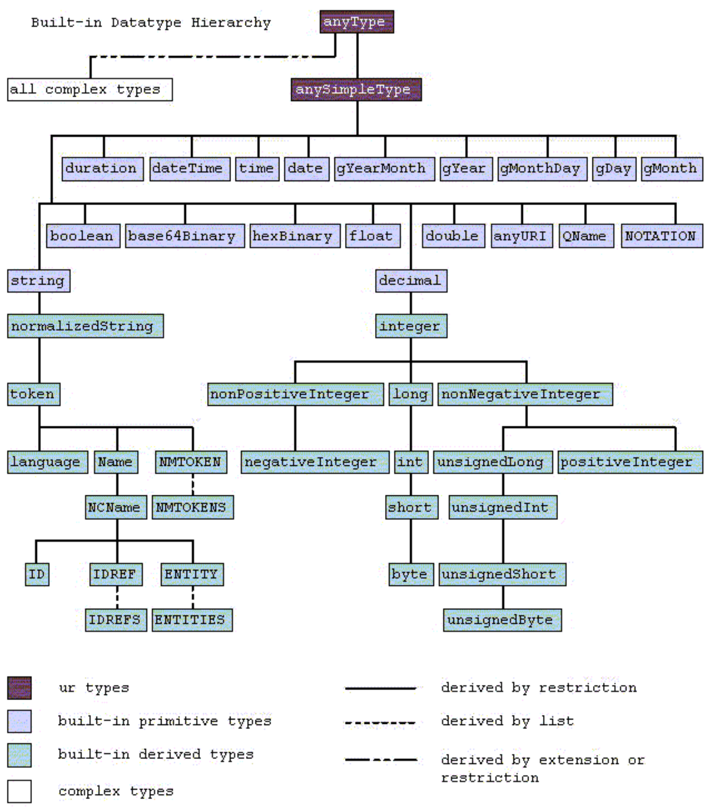

Figure 32. XML Schema built-in datatypes.

62 http://www.w3.org/TR/xmlschema-2/
63 http://www.w3.org/TR/rdf-schema/#ch_literal
64 http://www.w3.org/TR/rdf-concepts/#section-Literals

4.6.1 Data ADL (dADL) to OWL

<table>
<tr><th>ADL Leaf Data</th><th>W3C XML Schema and RDFS Literals</th></tr>
<tr><td>Character Data: <code>'a'</code></td><td>xsd:string</td></tr>
<tr><td>String Data: <code>"systolic blood pressure"</code></td><td>xsd:string</td></tr>
<tr><td>Integer Data: <code>3456</code></td><td>xsd:int</td></tr>
<tr><td>Real Data: <code>3.1415926</code></td><td>xsd:double</td></tr>
<tr><td>Boolean Data: <code>True, False</code></td><td>xsd:boolean</td></tr>
<tr><th colspan="2">Complete Date/Times</th></tr>
<tr><td>Date: <code>yyyy-MM-dd</code></td><td>xsd:date</td></tr>
<tr><td>Time: <code>hh:mm:ss[,sss][Z|+/-hhmm]</code></td><td>xsd:time</td></tr>
<tr><td>Date/Time: <code>yyyy-MM-ddThh:mm:ss[,sss][Z]</code></td><td>xsd:dateTime</td></tr>
<tr><th colspan="2">Partial Date/Times</th></tr>
<tr><td>Date with no days:
<code>yyyy-MM</code> and <code>yyyy-MM-??</code></td><td>xsd:gYearMonth</td></tr>
<tr><td>Time with no seconds:
<code>hh:mm and hh:mm:??</code></td><td>Translated as single integers[65]</td></tr>
<tr><td>Date/time with no seconds:
<code>yyyy-MM-ddThh:mm</code></td><td>Translated as single integers</td></tr>
<tr><td>Date/time, no minutes or seconds:
<code>yyyy-MM-ddThh</code></td><td>Translated as single integers</td></tr>
<tr><td>Date, no month or day:
<code>yyyy-??-?? and yyyy-??-?T??:??:??</code></td><td>xsd:gYear</td></tr>
<tr><td>Time, no minutes or seconds: <code>hh:??:??</code></td><td>Translated as a single integer</td></tr>
<tr><td>Date/time with no seconds:
<code>yyyy-MM-ddThh:mm:??</code></td><td>Translated as single integers</td></tr>
<tr><td>Date/time with no minutes or seconds:
<code>yyyy-MM-ddThh:??:??</code></td><td>Translated as single integers</td></tr>
<tr><th colspan="2">Intervals and Lists</th></tr>
<tr><td>Intervals of Ordered Primitive Types
Examples:
<code>|0..7|</code> <code>|0.0..5000.0|</code>
<code>|0.0..<5000.0|</code> <code>|02:13..05:10|</code>
<code>|>= 1984-02-05|</code> <code>|10.0 +/-10.0|</code>
<code>|>=0|</code></td><td>User-Defined Datatypes based on the xsp.owl ontology (see section 4.2)</td></tr>
<tr><td>Lists of Built-in Types
Examples:</td><td>Two approaches have been studied (see section 4.5):
- Enumerated Datatypes</td></tr>
</table>

[65] Not all partial Date/Times are supported by the W3C XML Schema

{"immobile", "very limited", "reduced", "full"} {"at0012", "at0013", "at0014"} {1, 4, 9, 16}	- *owl:hasValue* restrictions

Note: All dADL data eventually devolve to instances of the primitive types String, Integer, Real, Double, String, Character, various date/time types, lists or intervals of these types, and a few special types. The dADL representation do not use type or attribute names for instances of primitive types, only manifest values, making it possible to assume as little as possible about type names and structures of the primitive types.

4.6.2 Constraint ADL (cADL) to OWL

Set membership: `matches, is_in`	*owl:allValuesFrom* **and/or** *owl:hasValue* **restrictions**
Container Attributes	
Cardinality constraint: `cardinality`	*owl:cardinality*
Occurrences constraint: `occurrences`	*OWL 2.0 Qualified Cardinality Restrictions*[66]
"Any" constraint: `{*}`	Unrestricted OWL properties
Node Identifiers: `[at0123]`	*OWL Annotation Property* **named** *NodeID*
Internal References: `use_node`	*owl:allValuesFrom* restriction (see section 4.4)

[66] A workaround for OWL 1.0 implementations is described in section 4.3

5 The ADL2OWL translator implementation

The ADL to OWL translation principles described in Chapter 4 have been implemented in the *ADL2OWL* translator, which is a Java based and open source project. It includes, on one hand, a translation library composed of two packages (the *adl2owl.parser* and the *adl2owl.translator*), that can be invoked by third party applications to launch the translation process of a provided ADL archetype. On the other hand, the project includes a standalone application with a GUI (i.e. the *adl2owl.gui* package) that allows selecting an ADL file and translating it to an OWL representation. The GUI is based on the Swing toolkit (Loy & Eckstein, 2002), that provides a set of components for building GUIs and adding rich graphics functionality and interactivity to Java applications. The translator project has become part of the *openEHR Java Implementation Project*[67] and its source code is been integrated as a module[68].

A screenshot of the translator graphic interface is shown in Figure 33. There are radio buttons to select the language that is going to be used to name the resulting OWL classes and attributes that represent the archetype terms. Available languages are retrieved from the ontology section of the archetypes (see section 2.2.2), which

---

[67] http://www.openehr.org/projects/java.html
[68] http://www.openehr.org/svn/ref_impl_java/SANDBOX/ehr2ont/

provides a list for each language including human readable names and description of the constrained terms.

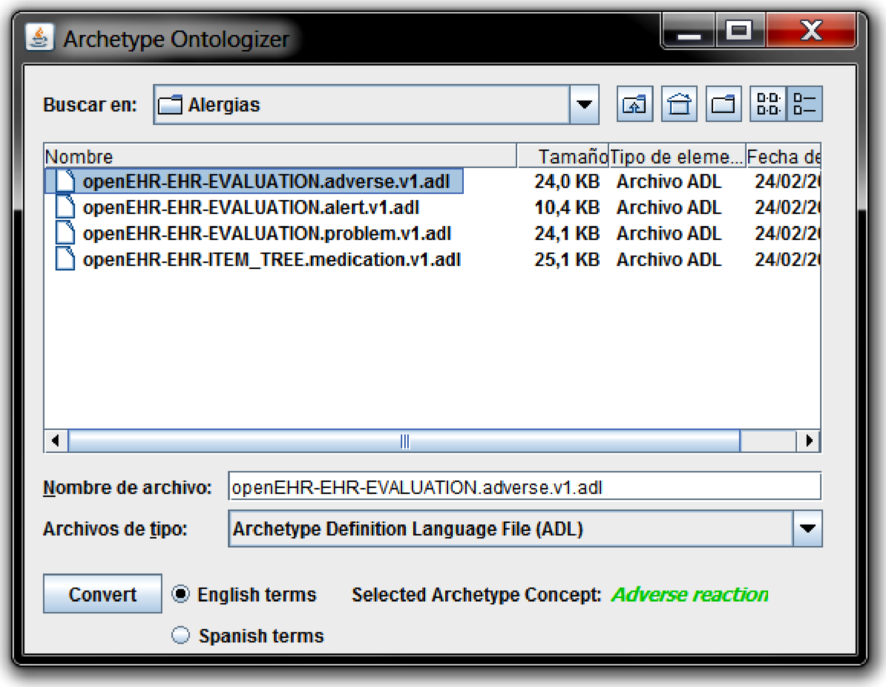

Figure 33. The ADL to OWL graphic user interface.

When the **Convert** button is pressed, the application loads the required RM ontologies (see section 4.1) and then performs the translation. An OWL file is generated containing the OWL representation of the archetype in the XML/RDF syntax. It should be noted that subsequent visualization and modifications of the OWL version of the archetype (e.g. with the Protégé editor) will continue requiring access to the RM ontologies in order to properly interpret the archetype.

The structure chart in Figure 34 shows the breakdown of the ADL2OWL functionality. This kind of chart, which has been typically used in structured programs design (Martin & McClure, 1985), allows understanding the sequence of subtasks that compose the translation mechanism, as well as their objectives. However, it should be noted that there is not a direct correspondence between these blocks and the source code distribution, which is based on the Object Oriented Programming paradigm. Therefore, a more precise description of each class implementation is provided in the next sections, including references to the structure chart subtask that they support.

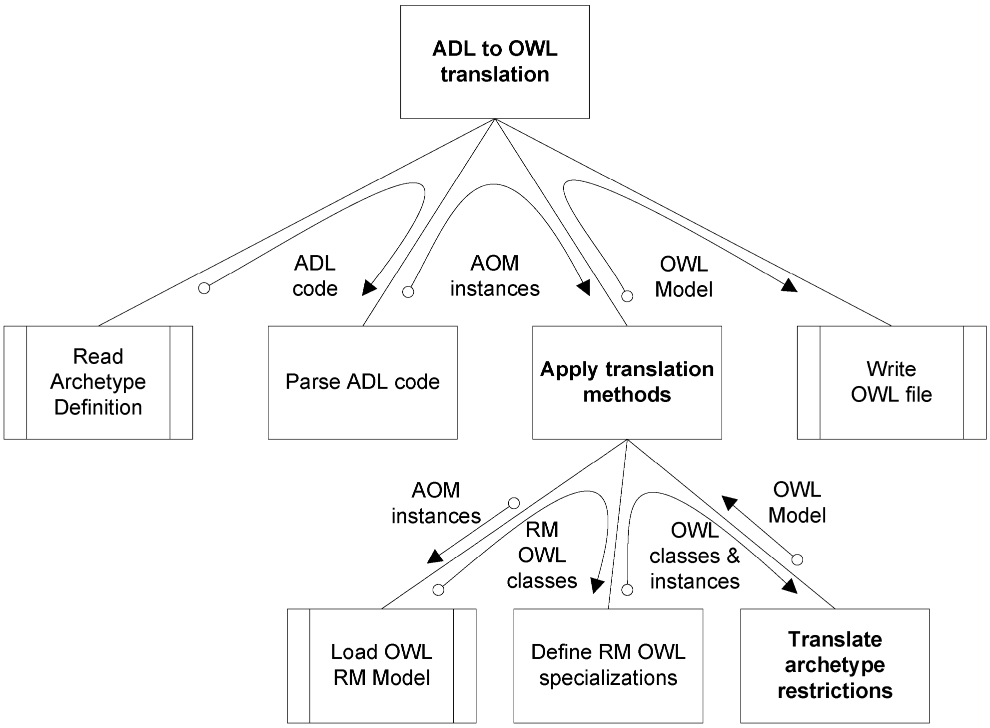

Figure 34. Structure chart of the ADL2OWL translator.

Figure 35 and Figure 36 are UML class diagrams describing the ADL2OWL java implementation packages. It should be noted that not all classes or dependencies in the implementation are described in this section. Only a subset of them is illustrated in order for the reader to understand the structure and design of the whole project. References and imports of the *edu.stanford.smi.protegex.owl* package as well as of the *org.openehr.am* and *org.openehr.rm* packages are quite common throughout the source code, but they are not represented in the following diagrams for the sake of simplicity. Significant classes are then explained one by one in sections 5.2 and 5.3. For *Translator* specialisations, particular translation techniques are described.

The *adl2owl.translator* package illustrated in Figure 36 is the main package within the ADL2OWL library. It contains the core implementations of the translation algorithms, without concerning about previous and post-translation issues like ADL parsing and saving the OWL file to disk, which are solved by the classes in the diagram of Figure 35. Such level of modularization allows the evolution of the translation algorithms with minimal modifications of the source code. New translator releases can be due to the growth of the OWL capabilities as a result of new OWL versions, providing a better representation mechanism, or simply because new keywords or constraints have been added to the ADL syntax and semantics.

Another design feature that considerably improves the implementation maintainability is that almost all classes in the *adl2owl.translator* package are organized in a common class hierarchy rooted by the *Translator* abstract class. The upper levels of the hierarchy include the source code for a few common steps like naming constraints (see section 5.3) that are shared by all translation algorithms. More specific decisions that depend on whether we are translating content, structure, representation or data constraints (i.e. the four subpackges within the *adl2owl.translator* package) are taken by polymorphic implementations in the lower levels of the hierarchy. In fact, the general translation context represented in the root of the hierarchy is not limited to openEHR ADL definitions. As a result of future work, the *OpenEHR2OwlTranslator* will be accompanied by other EHR translators addressing, for example, the CEN/ISO standard (see section 2.2).

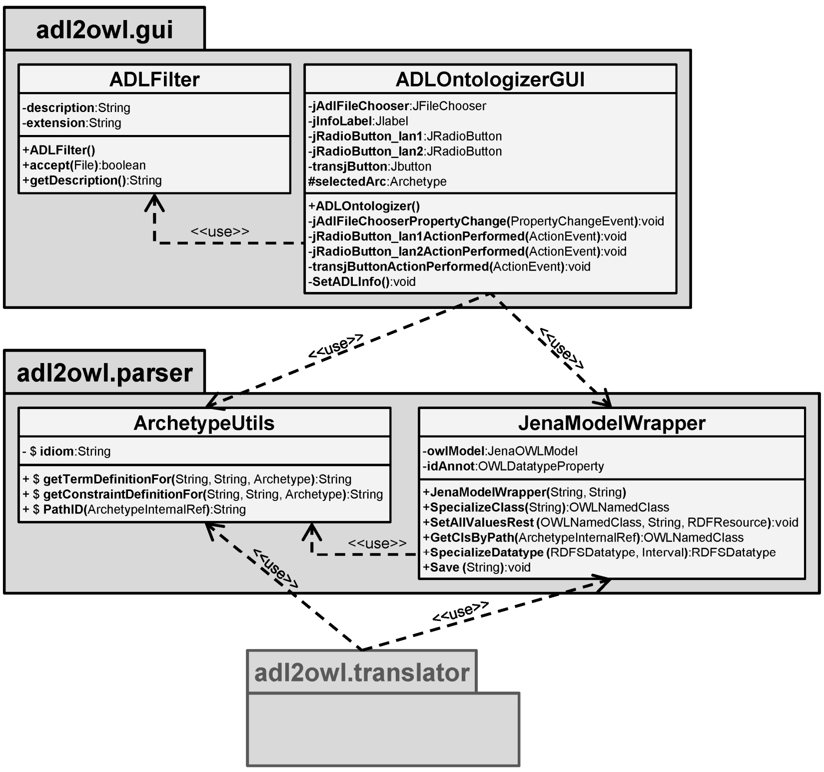

Figure 35. Dependencies between main packages.

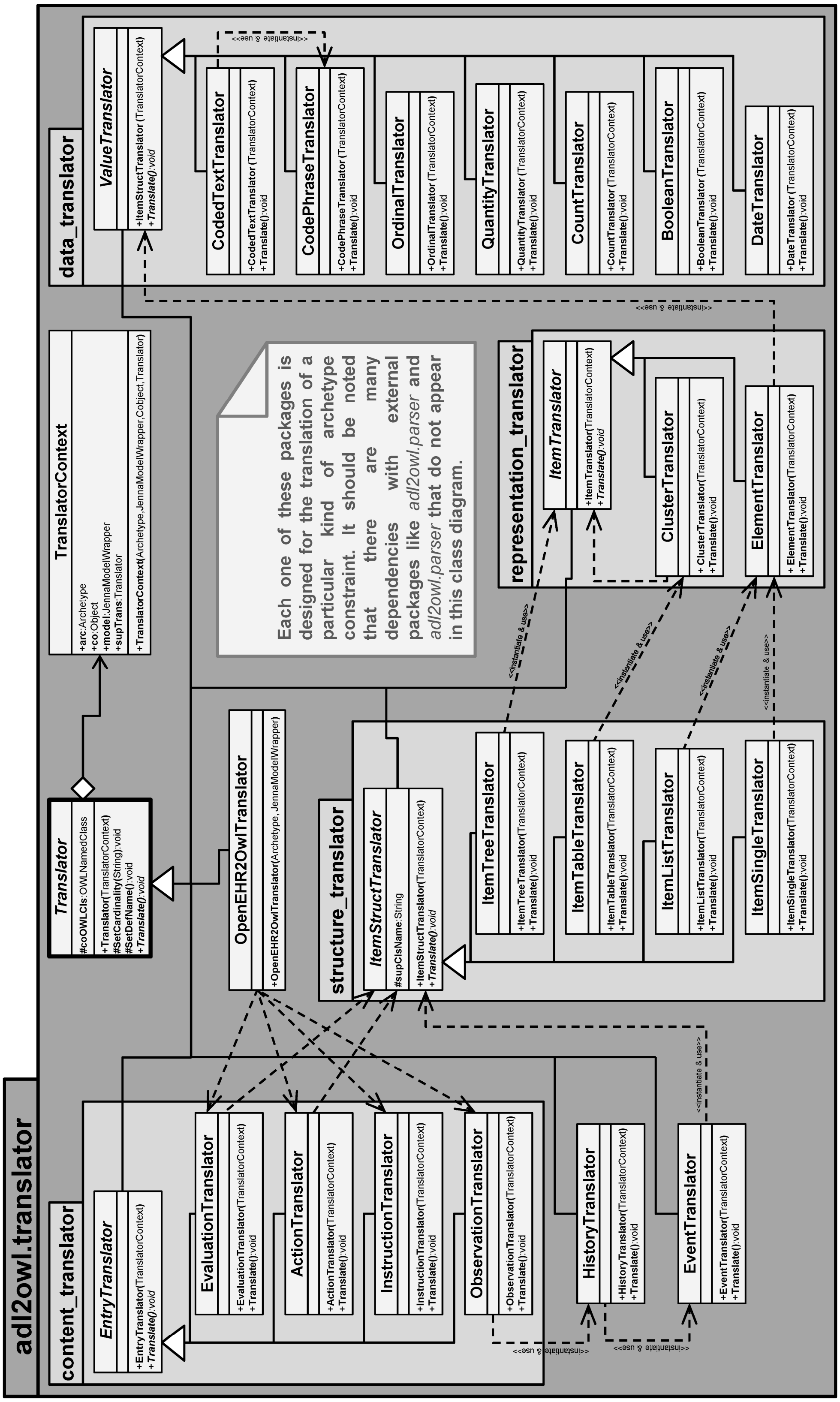

Figure 36. Class diagram of the translator package.

5.1 The graphic user interface package

This section provides the specifications for the java classes contained in the *adl2owl.gui* package shown in Figure 35.

CLASS	**ADLOntologizerGUI**	
Purpose	To define the user interface that allows selecting an ADL archetype and translating it to OWL. Although most archetypes are defined using English terms and names for the constrained clinical concepts, several languages may be available, according to the languages provided in the ontology section. It should be noted that the interface has been developed as a desktop application but it can be also implemented for the web environment.	
Inherit	*javax.swing.JFrame*	
Methods	**Signature**	**Specific description**
private	*void jAdlFileChooserPropertyChange (PropertyChangeEvent)*	This is the *FileChooser* handler. When an ADL file is highlighted, the archetype is parsed to extract the clinical concept metadata (it should be noted, for example, that the file name may not include the full clinical concept name) and the available languages that can be used to name the resulting OWL classes in the translation. All this information is show to the user.
private	*void jRadioButton_ActionPerformed (ActionEvent)*	This is the *radio buttons* handler, allowing the user to select the language that is going to be used when naming the OWL classes representing archetype elements.
private	*void transjButtonActionPerformed (ActionEvent)*	This is the *translation button* handler. It launches the translation process by invoking the root translation method (*OpenEHR2OwlTranslator* class) and loading the RM ontology.

CLASS	**ADLFilter**	
Purpose	This is a FileFilter implementation that has been set on the GUI File Chooser to keep non ADL files from appearing in the directory listing.	
Inherit	*javax.swing.filechooser.FileFilter*	
Methods	**Signature**	**Specific description**

public	*boolean accept(File)*	Implementation of the abstract method that checks the extension of the given file to *accept* it as an *Archetype Definition Language* file or not.

5.2 The parser package

This section provides the specifications for the java classes contained in the *adl2owl.parser* package shown in Figure 35.

CLASS	**ArchetypeUtils**	
Purpose	This class provide static methods to support frequent operations, related to archetype metadata and concept naming, identification and localization during the ADL to OWL translation	
Inherit	*javax.swing.filechooser.FileFilter*	
Methods	**Signature**	**Specific description**
public	*static String getTermDefinitionFor (String, String, Archetype)*	Traverses the ontology section of the given *archetype* in order to retrieve the readable name of the given ConceptID (*String*) in the given language (*String*).
public	*static String getTermDefinitionFor (String, String, Archetype)*	Traverses the ontology section of the given *archetype* in order to retrieve the referenced constraint (*String*) in the given language (*String*). This kind of constraint is described in the same archetype, but outside the main constraint structure. This is used to refer to constraints expressed in terms of external resources, such as constraints on terminology value sets.
public	*static String PathID (ArchetypeInternalRef)*	This method extracts the node ID (*String*) from the given archetype internal path (*ArchetypeInternalRef*).

CLASS	**JenaModelWrapper**
Purpose	This class encapsulates the original JenaOWLModel (*edu.stanford.smi.protegex.owl.jena*) to provide ad-hoc access to its functionality, according to ADL2OWL translation requirements. It should be noted that in an alternative implementation this class could *inherit* from JenaOWLModel to get benefits from protected resources and promote code reusability.

Inherit	*java.lang.Object*	
Methods	**Signature**	**Specific description**
public	*JenaModelWrapper (String, String)*	This is the constructor of the Jena wrapper. It loads the OWL file in the given path (*String*)[69], as well as the rest of context ontologies representing the RM classes that are going to be specialised according to the archetype constraints during the translation process.
public	*OWLNamedClass SpecializeClass(String)*	This method is invoked in order to create a clean specialisation of an RM class whose name is given by the parameter (*String*). It should be noted that this is a very frequent task during the translation process.
public	*void SetAllValuesRest (OWLNamedClass, String,RDFResource)*	This is an implementation of the *allValuesFrom* restriction that is used very frequently during translation (see section 4.1). The method creates an *UnNamedClass* that contains all instances fulfilling the given restriction (*String* and *RDFResource*), and then sets the *UnNamedClass* as a super class of the OWL class that is being restricted.
public	*OWLNamedClass GetClsByPath (ArchetypeInternalRef)*	Returns an OWL class which is the result of the previous translation of an archetype node or constraint given by *ArchetypeInternalRef*.
public	*RDFSDatatype SpecializeDatatype (RDFSDatatype, Interval)*	This is an implementation of the mechanism described in section 4.2 to restrain possible values of an archetype leaf node to the given datatype (*RDFSDatatype*) and the given *Interval*.
public	*void Save(String)*	Saves all the OWL classes, properties and restrictions that have been created as a result of the ADL2OWL translation. An OWL file is generated and stored in disk[52].

[69] The Protégé-OWL API can be used in two storage modes: the *OWL Files* mode and the *OWL Database* mode. The OWL Files mode, which is used to load and save the ontologies in this research, is based on the JenaOWLModel class. The static methods from the ProtegeOWL class are in integrated with Jena in order to provide these services. After receiving an existing OWL file from a stream or a URL, a JenaOWLModel is generated, that can be then used to write the file back to disk by means of its save methods.

5.3 The translator package

This section provides the specifications for the java classes contained in the *adl2owl.translator* package shown in Figure 36.

CLASS	***Translator (abstract)***	
Purpose	This is the root class of the translation hierarchy (see Figure 36), where each type of ADL constraint or data structure will have a dedicated specialisation of the *Translator* class. This abstract class includes attributes that will be used by all level translators as well as common functionalities that will be widely reused as the generalization of the OWL classes naming process.	
Inherit	*java.lang.Object*	
Methods	**Signature**	**Specific description**
public	*OWLNamedClass GetResult()*	Returns the OWL class which is result of the ADL node translation assigned to the given translator. If called before the execution of the *Translate()* returns *null*.
public	*OWLNamedClass GetContext()*	Returns a pack of instances that fully describe the context that must be considered by the translator.
protected	*void SetDefName()*	This method generalizes the naming process of the OWL specialisations resulting from translation, based on context information (i.e. the super class in the RM, the language selected by the user and the clinical concept description provided in the archetype ontology section)
protected	*void SetCardinality (String)*	If the RM attribute (*String*) connecting the RM specialisation that is being translated with the rest of the archetype is represented as an object of the *C_MULTIPLE_ATTRIBUTE* AOM class, then there may be an instance of the *CARDINALITY* AOM class attached to this attribute. Such class express constraints on the cardinality of container objects. In that case, the *SetCardinality()* method allows to translate such constraints to OWL, including *uniqueness* and *ordering*, providing the means to state that a container acts like a logical list, set or bag. It should be noted that the cardinality cannot contradict the cardinality of the corresponding attribute within the RM.
public	*abstract void Translate()*	This is simply the abstract definition of the *Translate()* method that will be further implemented according to the characteristics of each translator and

		the ADL node that it is designed for.

CLASS	**OpenEHR2OwlTranslator**	
Purpose	This class contains the *Translate()* method that is invoked from outside the package in order to start the ADL to OWL translation process. Therefore, it acts as the package interface with other packages and applications. For example, it is invoked from the ADL2OWL GUI (see section 5.1) where users can select the archetype that is going to be translated.	
Inherit	*translator.Translator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate()*	This method launches the translation of the archetype's root class, which is normally a specialisation of one of the following RM classes: `OBSERVATION`, `EVALUATION`, `INSTRUCTION` and `ACTION`. As the translation process is guided by a *DFS* traverse pattern (see section 4.1), the execution thread eventually returns to this method, when the whole archetype tree structure is translated. Finally the generated OWL classes are saved.

CLASS	**TranslatorContext**	
Purpose	Instances of this class gather all the attributes that compose the translations context for each ADL node.	
Inherit	*java.lang.Object*	
Attributes	**Signature**	**Specific description**
public	*Archetype arc*	Points to the input archetype object that will be translated
public	*Object co*	Points to the next node to be translated in the ADL hierarchy
public	*JennaModelWrapper model*	Points to an OWL ontology that will store the translation results.
public	*Translator supTrans*	Back pointer to the super ADL node translator.

CLASS	**HistoryTranslator**	
Purpose	To translate HISTORY specialisations to OWL, which are the roots of linear histories, like time series structures. For a periodic series of events, period will be set, and the time of each EVENT in the HISTORY must correspond. Missing events in a period HISTORY are however allowed.	
Inherit	*translator.Translator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate()*	The list of EVENTs is represented in OWL as an *unnamed union class*. Then relations through the events predicate are restricted to instances of this class, preventing any other EVENT from belonging to the current event series. The translation of each EVENT in the list is delegated to the EventTranslator class.

CLASS	**EventTranslator**	
Purpose	To translate EVENTs specialisations to OWL, defining the abstract notion of a single event in a series. EVENTs specialisations allow generic types of data structures, which are then locked to particular spatial types, e.g. EVENT<ITEM_LIST>. Specialisations express point or interval data.	
Inherit	*translator.Translator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate()*	The method determines the specific subtype of ITEM_STRUCTURE that is used inside this EVENT, in order to invoke the corresponding Translator (see section 5.3.2 about the *Structure Translator* package). Then relations through the data predicate for this EVENT will be restricted to the OWL representation of the ITEM_STRUCTURE specialisation provided by such translator.

5.3.1 The translator.content_translator package

CLASS	***EntryTranslator (abstract)***
Purpose	Includes general implementations for ENTRY subtypes translators. An ENTRY is the root of a logical item of "hard" clinical information created in the "clinical statement" context, within a clinical session. There can be numerous such contexts in a clinical session. For example

	OBSERVATIONs and other ENTRY subtypes only ever document information captured or created in the event documented by the enclosing COMPOSITION.	
Inherit	*translator.Translator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	All ENTRY subtypes must fulfil a protocol predicate that links the patient data to a description of how the information in this entry was arrived at. For OBSERVATIONs, this is a description of the method or instrument used. For EVALUATIONs, how the evaluation was arrived at. For INSTRUCTIONs, how to execute the instruction. Also, this may take the form of references to guidelines, knowledge references or clinical reasons within a larger care process. But in all cases the translation to OWL will guarantee this *description* to be represented by an instance of an ITEM_STRUCTURE specialisation.

CLASS	**EvaluationTranslator**	
Purpose	To translate EVALUATION specialisations to OWL, which are used for all kinds of statements which evaluate other information, such as interpretations of observations, diagnoses, differential diagnoses, hypotheses, risk assessments, goals and plans.	
Inherit	*content_translator.EntryTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	A mandatory and unique specialisation of ITEM_STRUCTURE is bound to the OWL representation of the archetype by applying an *owl:cardinality* and an *owl:allValuesFrom* restriction to the data RM attribute. Given that ITEM_STRUCTURE is an abstract class in the RM, archetype constraints are always defined over one of its subclasses: ITEM_TREE, ITEM_TABLE, ITEM_SINGLE or ITEM_LIST.

CLASS	**ObservationTranslator**
Purpose	To translate OBSERVATION specialisations to OWL, which are used to store all clinical data that has already occurred by the time it is recorded. OBSERVATION data is expressed using the class HISTORY, which guarantees that it is situated in time.

Inherit	*content_translator.EntryTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	An instance of a HISTORY specialisation must be linked to the archetype by the data RM attribute. This is guaranteed in the translation by an *owl:cardinality* and an *owl:allValuesFrom* restriction. In contrast, the value of the state RM attribute is optional for OBSERVATION specialisations, although it must also be restricted to HISTORY instances.

CLASS	**ActionTranslator**	
Purpose	To translate ACTION specialisations to OWL, which are used to record a clinical action that has been performed, which may have been ad-hoc, or due to the execution of an *activity* in an INSTRUCTION workflow. Every ACTION corresponds to a careflow step of some kind or another.	
Inherit	*content_translator.EntryTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	An *owl:allValuesFrom* restriction and an *owl:cardinality* one allow guaranteeing all ACTION specialisations to be linked to a subtype of ITEM_STRUCTURE through the description property. Such archetyped structure provides a description of the activity to be performed.

The InstructionTranslator, designed to translate INSTRUCTION specialisations to OWL, follows a similar mechanism to the one of ItemListTranslator.

5.3.2 The translator.structure_translator package

CLASS	***ItemStructTranslator (abstract)***
Purpose	Acts as a parent class in the ADL to OWL translation, providing common features for all spatial datatype translators as ItemTreeTranslator, ItemListTranslator, ItemSingleTranslator, ItemTableTranslator.
Inherit	*translator.Translator (abstract)*

CLASS	ItemTreeTranslator	
Purpose	To translate ITEM_TREE specialisations to OWL, which are commonly used to represent data which are logically a tree such as audiology results, microbiology results or biochemistry results.	
Inherit	*structuretranslator.ItemStructTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	A combination of an unnamed *OWLUnionClass* and an *owl:allValuesFrom* restriction is used to force the items RM attribute to target only to instances of the specialisations of CLUSTER or ELEMENT contained in the ITEM_TREE.

CLASS	ItemListTranslator	
Purpose	To translate ITEM_LIST specialisations to OWL, which are used to represent any data which is logically a list of values, such as blood tests, blood pressure, most protocols, etc.	
Inherit	*structuretranslator.ItemStructTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	A combination of an unnamed *OWLUnionClass* and an *owl:allValuesFrom* restriction is used to force the items RM attribute to target only to instances of the ELEMENT specialisations contained in the ITEM_LIST.

CLASS	ItemTableTranslator	
Purpose	To translate ITEM_TABLE specialisations to OWL, which are used to represent any data which is logically a table of values, such as *blood pressure*, most protocols, many *blood tests*, etc. Implemented using Cluster-per-row encoding. Each row CLUSTER must have an identical number of ELEMENTs, each of which in turn must have identical names and value types in the corresponding positions in each row.	
Inherit	*structuretranslator.ItemStructTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	Follows a similar mechanism to the one of ItemListTranslator, but restricting the rows property to CLUSTER specialisations.

CLASS	ItemSingleTranslator	
Purpose	To translate `ITEM_SINGLE` specialisations to OWL, which are used to represent any data which is logically a single value, such as a patient's height, weight or age.	
Inherit	*structuretranslator.ItemStructTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	An *owl:allValuesFrom* restriction forces the `item` RM attribute to target only to instances of the provided `ELEMENT` specialisation, while an *owl:cardinality* restriction guarantees a unique relation.

5.3.3 The translator.data_translator package

Translation mechanisms are similar when the translated constraint is also similar. For example similar kinds of quantities constraints are analogous in their translation. Therefore, this section includes the description of a subset of translators whose methods are unique in the subset. The rest of data translators are similar in their mechanisms to members of this subset.

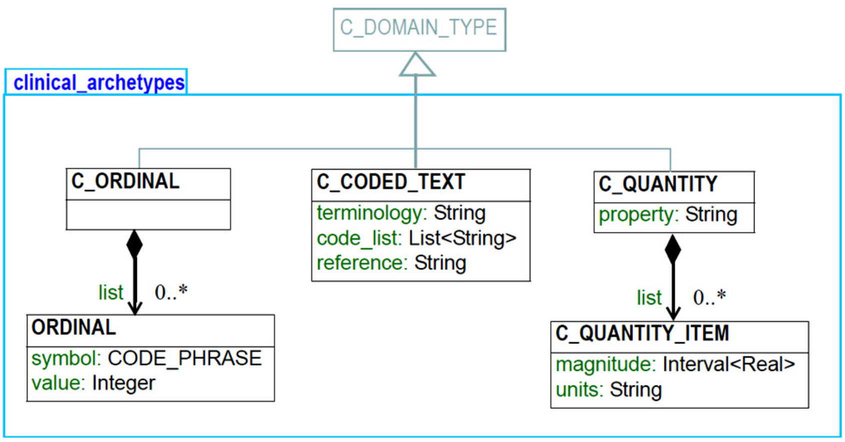

Figure 37. Example Domain-specific AOM classes.

As introduced in section 2.2.2, archetypes constraints are parsed and loaded as objects of the java AOM. Then the parsed nodes are linked by parent-children relations, building a hierarchical structure composed mainly of *CComplexObjects* (to represent RM types like `OBSERVATION`, `EVENT`, `ITEM_LIST`, etc) and *CAttributes* (to represent RM attributes like `data`, `events`, `items`, `value`, etc). This structure can be traversed by iterators. In addition, the AOM includes specific classes for each kind of data type constraint, for example, constraints on `ORDINAL`s are loaded as instances of the *C_DV_ORDINAL* AOM class, and in the same manner there is a `C_BOOLEAN` for `Boolean` values, *C_DATE_TIME* for `Time` values, etc.ws traversing archetype by iterators. Some of these classes are shown in Figure 37, taken from Beale (2007).

CLASS	**CodePhraseTranslator**	
Purpose	To translate instances of *C_CODE_PHRASE*, which are placed in archetypes leaf nodes in order to express constraints on instances of `CODE_PHRASE`. The `terminology_id` attribute may be specified on its own to indicate any term from a specified terminology; the `code_list` attribute may be used to limit the codes to a specific list.	
Inherit	*data_translator.ValueTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	Unnamed classes defined by *owl:hasValue* restrictions are combined with an O*WLUnionClass*, which is then used in an *owl:allValuesFrom* restriction to force the `code_list` RM attribute to target only to elements in a provided list of allowed codes (*Strings*). The list may be empty, meaning any code in the terminology may be used. The procedure is similar within the OrdinalTranslator. In addition, both translators may add human readable comments to the created OWL class describing the codes in one case and the ordinals in the other.

CLASS	**CodedTextTranslator**
Purpose	To translate `DV_CODED_TEXT` specialisations to OWL, which are used to represent rubrics from a controlled terminology. The translation to OWL guarantees a link with an instance of `CODE_PHRASE` through the `defining_code` RM attribute. The translation “hard work” is then delegated to the CodePhraseTranslator.
Inherit	*data_translator.ValueTranslator (abstract)*

<table>
<tr><th>CLASS</th><th colspan="2">QuantityTranslator</th></tr>
<tr><td>Purpose</td><td colspan="2">To translate instances of C_DV_QUANTITY, which are placed in archetypes leaf nodes that express constraints on instances of <code>DV_QUANTITY</code>. Although these quantities are typically used to represent “scientific” magnitudes and units, they can also be used for time durations, where it is more convenient to treat these as simply a number of seconds rather than days, months, years.</td></tr>
<tr><td>Inherit</td><td colspan="2">data_translator.ValueTranslator (abstract)</td></tr>
<tr><td>Methods</td><td>Signature</td><td>Specific translation technique</td></tr>
<tr><td>public</td><td>void Translate( )</td><td>This is one of the most complicated translation mechanisms as <code>DV_QUANTITY</code> constraints are defined as a set of allowed instances of C_QUANTITY_ITEM that can be described as triads (<code>magnitude</code>, <code>precision</code>, <code>units</code>) where the first is a range of <code>Double</code> values, the second is an interval of precisions to which the magnitude of the quantity must be expressed, in terms of number of decimal places (<code>Integer</code> interval), and the third is a <code>String</code> expressing the units in UCUM unit syntax. Both magnitude and precision constraints are user-defined datatypes.

The translation method is essentially a loop that takes as many iterations as triads are there in the C_DV_QUANTITY instance. Each iteration does the following:
i. Create an unnamed OWLIntersectionClass.
ii. Add the magnitude range to the intersection (see the SpecializeDatatype() method in section 5.2 for user-defined datatype creation).
iii. Add the precision interval to the intersection (same mechanism as in ii.).
iv. Create an unnamed OWLHasValue class with the units restriction and add it to the intersection.
v. Add the whole intersection as a member of an OWLUnionClass that will include all triads.

Finally the <code>DV_QUANTITY</code> specialisation is forced to take values among the resulting OWLUnionClass by means of an owl:allValuesFrom restriction.</td></tr>
</table>

CLASS	DateTranslator	
Purpose	To translate instances of *C_DATE* and *C_DATE_TIME*, which are placed in archetypes leaf nodes in order to express constraints on instances of `DV_DATE` and `DV_DATE_TIME` respectively. `DV_DATE` represents an absolute point in time, as measured on the Gregorian calendar, and specified only to the day, while `DV_DATE_TIME` is specified to the second.	
Inherit	*data_translator.ValueTranslator (abstract)*	
Methods	**Signature**	**Specific translation technique**
public	*void Translate( )*	The corresponding *xsd* datatype is chosen according to the date/time syntactic pattern defined in the archetype and retrieved by the AOM *getPattern()* method. A list of equivalences between ADL formats and XML Schema is given in section 4.6. However, it should be noted that there are some ADL date/time patterns like `yyyy-MM-ddThh` that does not have an *xsd* counterpart. In those cases, date/time values will be represented as a set of *xsd:integer* values and interpreted according to the given pattern. In addition, archetypes can further restrict the allowed date/time values by defining intervals or list of specific values. These restriction are retrieved by means of the AOM *getInterval()* and *getList()* methods, and translation to OWL follows similar mechanism to that of the *C_DV_QUANTITY* `magnitude` range translation (for the interval), and similar to the *C_CODE_PHRASE* `code_list` for lists of specific values.

5.3.4 The translator.representation_translator

CLASS	ElementTranslator
Purpose	To translate `ELEMENT` specialisations to OWL, which are simply the link between `DATA_STRUCTURE` specialisations (see section 5.3.2) and `DATA_VALUE` specialisations (see section 5.3.3). Therefore, the translation is delegated to the corresponding translator, according to the kind of value that is being constrained. `ELEMENT`s can include leaf nodes of the following types: `DV_PARAGRAPH`, `DV_URI`, `DV_STATE`, `DV_TEXT`, `DV_IDENTIFIER`, `DV_INTERVAL`, `DV_ORDERED`, `DV_BOOLEAN`, `DV_ENCAPSULATED`, `DV_TIME_SPECIFICATION`
Inherit	*representation_translator.ItemTranslator (abstract)*

CLASS	ClusterTranslator
Purpose	To translate CLUSTER specialisations to OWL, which constitute the grouping variant of ITEM (*abstract*). Thus, CLUSTER specialisations may contain instances of ELEMENTs and/or CLUSTERs, in an ordered list.
Inherit	*representation_translator.ItemTranslator (abstract)*

6 Integrating rules with clinical archetypes

Archetype definition languages (as the ADL specified by the OpenEHR consortium) currently support neither inference nor rules, so a first recommended step is to represent them as OWL ontologies that can be then enriched with clinical rules. The automatic mechanism of translation explained in chapters 4 and 5 allows developing and integrating archetypes into decision support and knowledge management software, as the archetype semantics are fully transferred to the ontology representation. Going a step further and incorporating rules to the OWL version of archetypes (e.g. SWRL rules) is an essential task toward the interoperability between heterogeneous systems.

In addition, sharing the knowledge expressed in the form of rules is coherent with the philosophy of open sharing underlying archetypes. While ontologies provide the basic framework for computational semantics, some inferential mechanisms allow reaching new conclusions that expand the boundaries of the declarative knowledge encoded in archetyped data. This is useful, for example, to support the application of knowledge about procedures which is typically contained in clinical guidelines. Besides, having inference mechanisms and descriptive knowledge combined under the same syntactic structure provides means for the interoperability of rule systems. As pointed out by Horrocks, Patel-Schneider, Bechhofer, and Tsarkov (2005), the integration of the OWL ontologies and SWRL offers several advantages and goes beyond that of either OWL DL or Horn rules alone.

This chapter is structured as follows: section 6.1 describes the rationale for sharing SWRL rules in the context of archetype-based ontologies. Then section 6.2 explains in four subsections the *ArchOnt* framework approach to archetype, rules and inference integration. The explanation is illustrated with a case study designed to improve patient safety for a particular situation, i.e. to provide decision support for *Intraoperative Monitoring and Prevention of Complications during Ligation of the Sigmoid or Transverse Sinus*. It should be noted that medical approval must be requested if the clinical guidelines referenced in this thesis to illustrate the representation and inference mechanisms will be used in new real scenarios.

6.1 Sharing and Reusing SWRL rules

As SWRL rules are defined, the need for a sharing method appears. Rule storage is not a problem because it is already implemented how to represent them as OWL individuals, and they can be stored using Semantic Web frameworks for data management as HP Jena[70]. As explained by O'Connor et al. (2005), SWRL rules are described by the SWRL Ontology[71]. However we must consider the case where the OWL version of an archetype and the SWRL rule associated with it are stored in separated files. This supports archetype translation and rule definition to be carried out at different times or by different groups of experts.

It should be noted that this is only a storage/management issue and it really does not make sense to semantically separate rules from ontologies. Like standard OWL axioms, SWRL rules are not disembodied entities so they can be interpreted only in terms of the ontology that they refer to. In order to preserve the semantic link, the following procedure is accomplished. Every time the same archetype is translated, an identical and unique URI is assigned to its OWL version. The URI generation is based on the archetype name and version. By applying the same generation mechanism, SWRL rules can safely reference to their archetypes. Thereby, whenever they are imported in the same OWL project they will connect properly.

70 http://jena.sourceforge.net/

71 http://www.w3.org/Submission/SWRL/swrl.owl

Modelling knowledge by using SWRL rules and then sharing them by means of SWRL repositories can be considered practices consistent with the sharing principles of archetypes. It should be noted that SWRL rules based on openEHR archetypes can be hosted together with archetypes or in their own *Clinical Rules Repository*, as illustrated in Figure 38. Reasons for sharing and reusing SWRL rules include the ones listed in what follows:

- **Interoperable decision support**: The ability of systems to reliably communicate with each other regarding clinical decision support. To encourage the development of interoperable mechanisms for triggering critical aids to decision making like alerts, reminders and monitoring tasks that improve effectiveness and reduce clinical risks.
- **Inheritable compatibility**: Given the archetypes' capability of being defined as specialisations of more general archetypes, a SWRL rule originally designed according to the OWL version of a parent archetype is also applicable to derived archetypes.
- **Fostering semantics for clinical guidelines**: The introduction of SWRL rules and inferential mechanisms together with the archetypes expand the boundaries of the declarative knowledge that can be migrated from clinical guidelines to healthcare information systems. In this manner, a means for standardized representation, reuse and execution of the essential fragments of declarative knowledge contained in clinical guidelines is provided.
- **Specialists' empowerment**: To enable domain rules and guidelines to be modelled in a formal way by domain experts. By defining the declarative knowledge they work with, they can gain direct control over their information systems.
- **Consistency checking**: Rules integration can offer consistency checks to help guaranteeing data correctness in EHR fragments.
- **Archetype validation**: To support archetype validation and inconsistent restrictions detection, according to the RM or the specialised parent archetype.
- **Full semantic interoperability**: For all above mentioned reasons, integrating rules with clinical archetypes and EHR is an essential step towards level 3 of SIOp.

This has the additional potential benefit of being complemented with inferential models defined on some clinical ontologies (not coming from archetype translation) as will be described in the case studies of Chapter 8.

6.2 Rules integration and inference execution

The four subsections of this section will explain the information and knowledge workflow from archetypes' translation to inference execution. The process is illustrated in Figure 38 and it is essentially a concatenation of the four white boxes that compose the *ArchOnt* Framework.

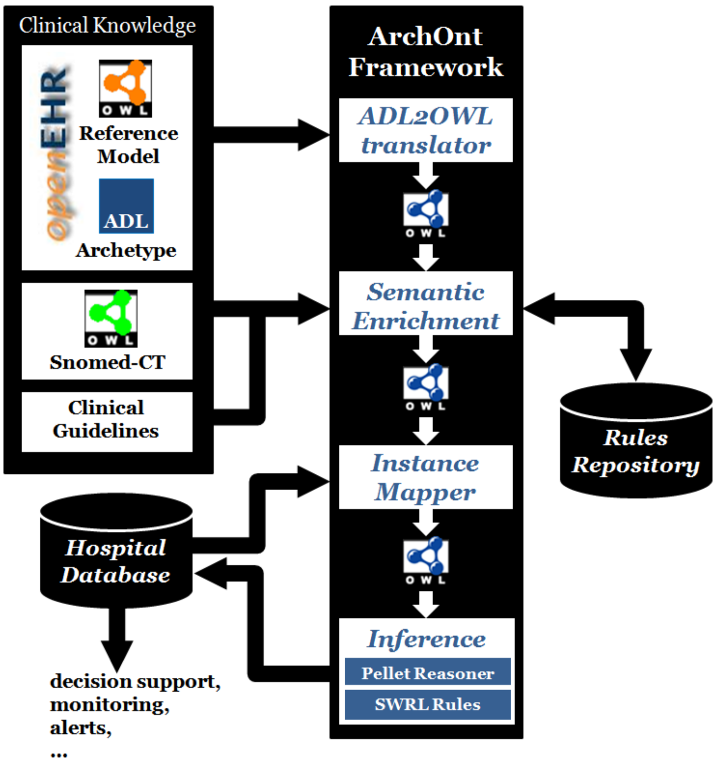

Figure 38. The ArchOnt Framework.

The first step is to translate the involved openEHR archetypes to OWL by combining their ADL definitions with the RM already expressed in OWL (*ADL2OWL translator* box). Then the resulting ontology is enriched with bindings to SNOMED-CT terms and clinical rules (*Semantic Enrichment* box). Once that all the required clinical knowledge is represented in a common OWL context, the archetypes are instantiated with patients' data coming from hospitals databases (*Instance Mapper* box). Finally, a semantic reasoner is invoked to gather the results from inference and feedback the hospitals' information systems (*Inference* box).

To support the framework explanation, each step will be described by means of an example based on the *Guidelines for the Ligation of the Sigmoid or Transverse Sinus during Large Petroclival Meningioma Surgery* (Hwang et al., 2004), supported by the Seoul National University Hospital the Korea Brain and Spinal Cord Research Foundation. In order to properly understand the role that archetypes and SWRL rules can play in this scenario, a brief description of the guideline and its key concepts is given below.

Transverse Sinuses are two areas within a human head and beneath the brain, which allow blood veins to span the area, from the back of the head towards the nose. *Sigmoid Sinuses* are also two areas beneath the brain, which allow blood veins to span from the centre of the head downward. A *Petroclival Meningioma* is a type of brain tumour located near the skull base, in an area known as the *petroclival junction*. Such tumours are challenging to treat, as they are located deep inside the brain and may be difficult to access surgically. Integral to the surgery application are the manipulation, management, and sacrifice of the transverse or sigmoid sinus, as total resection of large tumours requires ligation and resection of the sinus to obtain a sufficiently wide exposure of the tumour.

So, one of the moments where decision support is quite useful is while assessing whether sinuses can be ligated or not, as ligation under inappropriate circumstances can take to venous complications. The guideline section titled "*Intraoperative Monitoring and Prevention of Complications*" includes two references to straightforward recommendations on how to avoid such complications:

- The first one involves measuring **sigmoid sinus pressure** after test clamping of the sinus to assess contralateral venous drainage before cutting the sinus (Spetzler, Daspit, & Pappas, 1992). If pressure in the sigmoid sinus increases more than **10 mm Hg** by temporary occlusion testing, the sinus should be kept intact.
- The second one is provided by Day, Fukushima, and Giannotta (1997), founding that **intravascular pressure no greater than 5 mm H2O** was safe during test clamping. It should be noted that different units are

used in each case to represent intravascular pressure, mercury (Hg) and water (H2O).

6.2.1 Translating involved archetypes

Integrating the above introduced clinical knowledge with a reasoner requires representing intravascular pressure measures before and after temporary occlusion testing of the sinuses. Besides, it should be a widely accepted representation in order to foster rules interoperability between heterogeneous healthcare systems. The *openEHR-EHR-OBSERVATION.intravascular_pressure* serves this purpose as it supports pressure measures, allowing both units (Hg and H2O), as well as related information like location and device. The full ADL file is shown below. In this case, the SWRL rule will retrieve data from this archetype only, but there is no limit on the number of archetypes that can be accessed from a single rule. In fact, richer results come from processing several sources as illustrated in the case studies of Chapter 8.

```
archetype (adl_version=1.4)
  openEHR-EHR-OBSERVATION.intravascular_pressure.v1

concept
  [at0000]  -- Intravascular pressure
language
  original_language = <[ISO_639-1::en]>
description
  original_author = <
    ["name"] = <"Sam Heard">
    ["organisation"] = <"Ocean Informatics">
    ["date"] = <"28/06/2006">
    ["email"] = <"sam.heard@oceaninformatics.biz">
  >
  details = <
    ["en"] = <
      language = <[ISO_639-1::en]>
      purpose = <"Intravascular venous, arterial, pulmonary or cardiac pressure measurement">
      use = <"">
      keywords = <"pressure", "intravascular">
      misuse = <"Not to be used for systemic blood pressure. Use 'observation.blood_pressure'
               for this.">
    >
  >
  lifecycle_state = <>
  other_contributors = <>

definition
  OBSERVATION[at0000] matches {  -- Intravascular pressure
    data matches {
      HISTORY[at0001] matches {  -- history
        events cardinality matches {1..*; unordered} matches {
          EVENT[at0002] occurrences matches {0..*} matches {  -- Any event
            data matches {
              ITEM_TREE[at0003] matches {  -- Tree
                items cardinality matches {0..1; ordered} matches {
                  ELEMENT[at0005] occurrences matches {0..1} matches {  -- Pressure
                    value matches {
                      C_DV_QUANTITY <
                        property = <[openehr::125]>
                        list = <
                          ["1"] = <
                            units = <"mm[Hg]">
```

```
                            magnitude = <|>=0.0|>
                            precision = <|2|>
                          >
                          ["2"] = <
                            units = <"cm[H20]">
                            magnitude = <|>=0.0|>
                            precision = <|2|>
                          >
                        >
                      >
                    }
                  }
                  ELEMENT[at0015] occurrences matches {0..1} matches {  -- Relative pressure
                    value matches {
                      0|[local::at0016],   -- markedly reduced
                      2|[local::at0017],   -- lowered
                      4|[local::at0018],   -- normal/expected
                      6|[local::at0019],   -- raised
                      8|[local::at0020]    -- markedly increased
                    }
                  }
                  ELEMENT[at0006] occurrences matches {0..1} matches {  -- Location
                    value matches {
                      DV_CODED_TEXT matches {
                        defining_code matches {[ac0001]}  -- Any term that is a major blood
                                                          -- vessel or heart cavity
                      }
                    }
                  }
                  ELEMENT[at0007] occurrences matches {0..1} matches { --Phase of heart cycle
                    value matches {
                      DV_CODED_TEXT matches {
                        defining_code matches {
                          [local::
                          at0008,   -- Systolic
                          at0009,   -- Diastolic
                          at0023,   -- Pre-systolic
                          at0024]  -- Pre-diastolic
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
    protocol matches {
      ITEM_LIST[at0021] matches {  -- List
        items cardinality matches {0..*; unordered} matches {
          ELEMENT[at0022] occurrences matches {0..1} matches {  -- Device
            value matches {
              DV_TEXT matches {*}
            }
          }
        }
      }
    }
  }

ontology
  term_definitions = <
    ["en"] = <
      items = <
        ["at0000"] = <
          description = <"The pressure in a specific location, blood vessel or heart cavity,
                         at a specific phase of the heart or an average over the heart
                         cycle.">
          text = <"Intravascular pressure">
        >
        ["at0001"] = <
          description = <"@ internal @">
          text = <"history">
        >
        ["at0002"] = <
          description = <"Generic event">
          text = <"Any event">
        >
        ["at0003"] = <
          description = <"@ internal @">
```

```
            text = <"Tree">
          >
          ["at0005"] = <
            description = <"The pressure measured">
            text = <"Pressure">
          >
          ["at0006"] = <
            description = <"The location of the pressure measurement">
            text = <"Location">
          >
          ["at0007"] = <
            description = <"The phase of the heart cycle at the time of the measurement">
            text = <"Phase of heart cycle">
          >
          ["at0008"] = <
            description = <"During contraction of the heart">
            text = <"Systolic">
          >
          ["at0009"] = <
            description = <"During relaxation of the heart">
            text = <"Diastolic">
          >
          ["at0015"] = <
            description = <"The pressure in relative terms">
            text = <"Relative pressure">
          >
          ["at0016"] = <
            description = <"The pressure is much lower than normal or expected">
            text = <"markedly reduced">
          >
          ["at0017"] = <
            description = <"The pressure is reduced">
            text = <"lowered">
          >
          ["at0018"] = <
            description = <"The pressure is normal or as expected">
            text = <"normal/expected">
          >
          ["at0019"] = <
            description = <"The pressure is raised">
            text = <"raised">
          >
          ["at0020"] = <
            description = <"The pressure is much higher than normal or expected">
            text = <"markedly increased">
          >
          ["at0021"] = <
            description = <"@ internal @">
            text = <"List">
          >
          ["at0022"] = <
            description = <"The device used to measure the pressure">
            text = <"Device">
          >
          ["at0023"] = <
            description = <"Phase of the heart immediately prior to contraction of the heart">
            text = <"Pre-systolic">
          >
          ["at0024"] = <
            description = <"The phase of the heart immediately prior to filling of the
                          ventricle">
            text = <"Pre-diastolic">
          >
        >
      >
    >
    constraint_definitions = <
      ["en"] = <
        items = <
          ["ac0001"] = <
            description = <"A term that describes the location of the pressure measurement">
            text = <"Any term that is a major blood vessel or heart cavity">
          >
        >
      >
    >
```

In the first stage, illustrated as the *ADL2OWL translator* box in Figure 38, the *Intravascular Pressure* archetype is translated to OWL according to the translation principles in Chapter 4 and the implementation described in Chapter 5. Some of the translated nodes are shown in Figure 39.

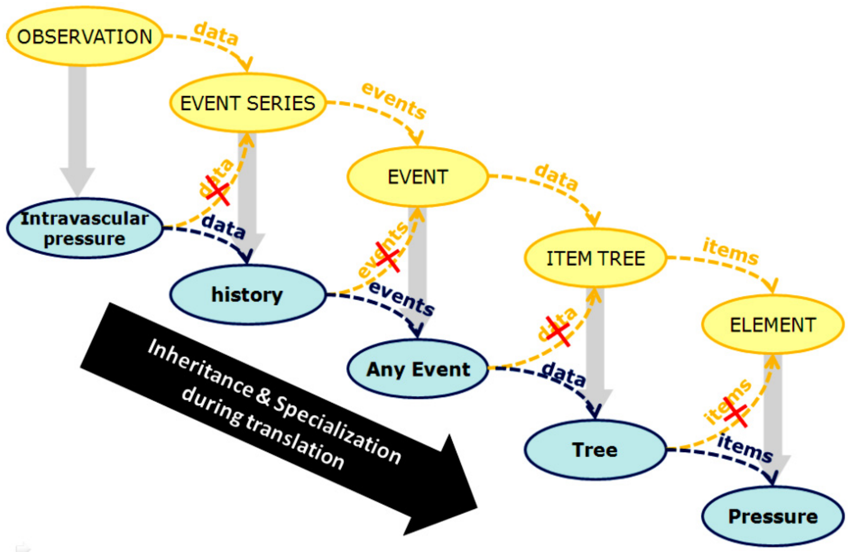

Figure 39. ADL to OWL translation of the *Intravascular Pressure* archetype.

6.2.2 Defining *linker rules* and *guideline rules*

When the archetype is represented in OWL, SWRL rules with above clinical guidelines recommendations can be attached. Two kinds of rules with different goals can be devised. First, there are rules that search for useful data in the archetype and associate it to the *Patient* class. This kind of rule will be called *linker rule*. Given that archetype tree structures can get very dense and deep when describing some concepts, linker rules help to quickly access relevant information, and therefore make other rules simpler. It should be noted that relevant information depends on the clinical guideline that is been represented in each case. In the *Sinus Ligation* example under study relevant data includes pressure values, location, and units of measure.

$$
\begin{aligned}
Patient(?pat) &\wedge IntravascularPressure(?ipr) \wedge \\
tests(?pat, ?ipr) &\wedge List(?lis) \wedge \\
protocol(?ipr, lis?) &\wedge Device(?dev) \wedge \\
itmes(?lis, ?dev) &\wedge I\_DV\_Text(?txt) \wedge \\
value(?dev, ?txt) &\wedge value(?txt, \text{“BEFORE Test Clamping”}) \wedge \\
History(?his) &\wedge data(?ipr, ?his) \wedge \\
Any\_event(?evt) &\wedge events(?his, ?evt) \wedge \\
Tree(?tre) &\wedge data(?evt, ?tre) \wedge \\
Location(?loc) &\wedge items(?tre, ?loc) \wedge \\
Location\_DV\_CODED\_TEXT(?ctx) &\wedge defining\_code(?ctx, \text{“279264003”}) \wedge \\
Pressure(?prs) &\wedge itmes(?tre, ?prs) \wedge \\
Pressure\_DV\_QUANTITY(?iqt) &\wedge value(?prs, ?iqt) \\
\Longrightarrow & \\
& hasBeforeTCpressure(?pat, ?iqt)
\end{aligned}
$$

Figure 40. Monitoring *Sinus Ligation* (SWRL context preparation I).

Figure 40 and Figure 41 contain two *linker rules* that connect *Patient* instances directly to pressure measures from before and during test clamping. In order to reach data, both rules traverse the archetype from containers to leaf nodes. A brief description about SWRL syntax and operations is given in section 2.3.2. The rule in Figure 40 checks that the measure has been taken in the *Sigmoid Sinus* (represented by it SNOMED code “279264003”), and “BEFORE Test Clamping”. Then it sets the direct link *hasBeforeTCpressure* (TC –> Test Clamping) between the patient and the pressure value. The rule in Figure 41 proceeds analogously but checking if the measure has been taken during test clamping and setting the *hasDuringTCpressure* predicate.

$$
\begin{aligned}
Patient(?pat) &\wedge IntravascularPressure(?ipr) \wedge \\
tests(?pat, ?ipr) &\wedge List(?lis) \wedge \\
protocol(?ipr, lis?) &\wedge Device(?dev) \wedge \\
itmes(?lis, ?dev) &\wedge I\_DV\_Text(?txt) \wedge \\
value(?dev, ?txt) &\wedge value(?txt, \text{“DURING Test Clamping”}) \wedge \\
History(?his) &\wedge data(?ipr, ?his) \wedge \\
Any\_event(?evt) &\wedge events(?his, ?evt) \wedge \\
Tree(?tre) &\wedge data(?evt, ?tre) \wedge \\
Location(?loc) &\wedge items(?tre, ?loc) \wedge \\
Location\_DV\_CODED\_TEXT(?ctx) &\wedge defining\_code(?ctx, \text{“279264003”}) \wedge \\
Pressure(?prs) &\wedge itmes(?tre, ?prs) \wedge \\
Pressure\_DV\_QUANTITY(?iqt) &\wedge value(?prs, ?iqt) \\
\Longrightarrow & \\
& hasDuringTCpressure(?pat, ?iqt)
\end{aligned}
$$

Figure 41. Monitoring *Sinus Ligation* (SWRL context preparation II).

Now we are in place to define the SWRL rule that actually captures the Spetzler et al. (1992) recommendation. This kind of rule will be called *guideline rule*. It is shown in Figure 42, where the *subtract* and *greaterThan* mathematical built-ins allow asserting an "inadequate" ligation.

$$
\begin{array}{rl}
Patient(?pat) & \wedge \; Pressure\_DV\_QUANTITY(?btValue) \wedge \\
hasBeforeTCpressure(?pat, ?btValue) & \wedge \; units(?btValue, \text{"mm[Hg]"}) \wedge \\
Pressure\_DV\_QUANTITY(?drValue) & \wedge \; hasDuringTCpressure(?pat, ?drValue)) \wedge \\
units(?drValue, \text{"mm[Hg]"}) & \wedge \; swrlb : subtract(?result, ?drValue, ?btValue) \wedge \\
swrlb : greaterThan(?result, 10) & \\
\Longrightarrow & \\
 & hasLigationAssessment(?pat, \text{"inadequate"})
\end{array}
$$

Figure 42. SWRL representation of the assessment of *Sinus Ligation*.

The semantics of the above presented approach to clinical guideline representation heavily relies on the combination of *linker rules* and *guideline rules*. The *Intravascular Pressure* archetype provides significant information like *Location* and *Device*, but they are not enough to completely define the *Sinus Ligation* monitoring context that is needed. Currently, such gap is filled by *linker rules*. However, an alternative approach could benefit from the archetypes' capability to be defined as specialisations of a parent and more general archetype. A solution can be then the creation of the *Intravascular Test Clamping* archetype as a specialisation of *Intravascular Pressure* including specific concepts like *Prior to test occlusion pressure* and *During test occlusion pressure*. In this sense, the equilibrium should be found between the specificity of archetypes and the size and complexity of SWRL rules. Then a similar procedure can be carried out for the Day et al. (1997) recommendation.

6.2.3 Mapping patients' data to OWL instances

For the inference to generate results, the OWL version of archetypes must be first provided with patient's data. Concrete clinical data may come from a variety of sources, e.g. relational databases or flat files. Here the general approach of translating scalar data from comma separated value (CSV) files into ontology instances is described. It should be noted that a similar procedure can be followed if the data is provided by a relational database.

Archetype instantiation involves the creation of a set of OWL individuals that conforms the archetype hierarchy. Because upper level instances cannot a priori establish which lower level instances will be supplied, the latter ones should be created first in order to properly fulfil the properties between levels. Thus, instantiation process performs a bottom-up traverse of the archetype.

Leaf nodes instances are concrete clinical values, originally stored outside the boundaries of OWL domain, for example, in a CSV file. For that reason, a mapping between those values and their archetype slots should be provided. Once the source values are transformed into OWL customized Datatypes, they are linked to `ELEMENT` instances, which are organized by an `ITEM_STRUCTURE` instance and so on, until the archetype's root instance is created.

The following CSV fragment contains an extract of clinical measures required to instantiate the *Intravascular Pressure* archetype. Each row is parsed and transformed into a set of OWL individuals, assembling an *Intravascular Pressure* instance.

```
Patient_ID | Date | Pressure | Relative | Location | Device
846385 | 09-03-2007 | 0.15 mmHg | normal | 279264003 | BEFORE Test Clamping
846385 | 09-03-2007 | 0.40 mmHg | raised | 279264003 | DURING Test Clamping
973589 | 20-10-2007 | 0.25 mmHg | normal |279264003 | BEFORE Test Clamping
973589 | 20-10-2007 | 10.45 mmHg | markedly increased |279264003 | DURING Test Clamping
123453 | 12-02-2008 | 0.22 mmHg | normal | 279264003 | BEFORE Test Clamping
123453 | 12-02-2008 | 0.30 mmHg | normal | 279264003 | DURING Test Clamping
745385 | 15-05-2010 | 0.19 mmHg | normal | 279264003 | BEFORE Test Clamping
745385 | 15-05-2010 | 13.7 mmHg | markedly increased | 279264003 | DURING Test Clamping
984647 | 03-07-2010 | 0.14 mmHg | normal | 279264003 | BEFORE Test Clamping
984647 | 03-07-2010 | 0.48 mmHg | raised | 279264003 | DURING Test Clamping
```

Mappings are node IDs of the current archetype ordered the same way as CSV fields:

```
 |  | at0005 | at0015 |at0006 | at0022
```

The lowest constraints in the archetype tree structure having a node ID are the `ELEMENT`s. Therefore, `at0005` indicates a *Pressure* `ELEMENT` value, `at0015` a *Relative Pressure* `ELEMENT`, and so forth. Neither *Patient_ID* nor *Date* fields have mappings to the archetype because such information is not described as a part of *Intravascular Pressure* concept. Actually, it is administrative information that should be captured by other archetypes. Assuming that CSV attributes notation is compatible with archetype slots or that it can be made compatible by an automatic transformation, the above procedure can be generalized to any kind of archetype.

Hospitals and healthcare centres sometimes provide very large files of data like these, including measures and results of thousands of tests and patients. So many rows of data can generate huge ontologies that compromise the efficiency and performance of the semantic reasoner. In fact, analysing and improving the performance and scalability of OWL reasoners is an ongoing challenge (Bail, Parsia, & Sattler, 2010). Under the context described above and considering clinical guidelines particular characteristics, several scalability tests were carried out. It has been found that the best performance is reached when not all patients' records are simultaneously loaded in the knowledge base. Instead, the whole inference cycle is execute over a set of 50 or 100 records, and then executed again over the next 50 or 100 records, until data is over. It should be noted that the fragmentation is feasible because records do not depend on each other. Such straightforward fragmentation approach cannot be done for example when loading the OWL version SNOMED-CT.

6.2.4 SWRL edition and execution

Two implementations have been explored that support then edition and execution of above defined clinical rules. Firstly, the capabilities of Protégé 3.4 that includes a SWRL editor have been tested in combination with the Jess[72] engine. The Jess engine is a production rule system with forward chaining to achieve inferred results. Mei and Paslaru (2005) have demonstrated that mappings between SWRL and Jess are possible. An alternative and newer approach is the one provided by Protégé 4 in combination with Pellet[73]. Although satisfactory results were obtained from both approaches, each one has his own capabilities and differences which are pointed out below.

The SWRL Tab

The SWRLTab[74] was developed for working with SWRL rules in Protégé. It is part of Protégé-OWL 3.4 and does not need to be downloaded separately. It provides a set of libraries that can be used in rules, including libraries to interoperate with XML documents, and spreadsheets, and libraries with mathematical, string, RDFS,

[72] http://www.jessrules.com/
[73] http://clarkparsia.com/pellet/
[74] http://protegewiki.stanford.edu/wiki/SWRLTab

and temporal operators. The SWRL Tab includes the following software components:

- *SWRL Editor:* Supports editing and saving of SWRL rules in an OWL ontology.
- *SWRL APIs:* The SWRLTab provides a collection of Java APIs to work with SWRL rules.
- *SWRL Built-in Libraries:* A number of built-in libraries are provided by the SWRLTab. These include an implementation of the core SWRL built-ins defined in the SWRL Submission (see section 2.3.2) and built-ins for querying OWL ontologies.
- *SWRL Built-in Bridge:* SWRL built-ins are user-defined predicates that can be used in SWRL rules. The SWRLTab has a subcomponent called the built-in bridge that provides a mechanism to define Java implementations of SWRL built-ins. These implementations can then be dynamically loaded by the bridge and invoked from a rule engine.
- *SWRL Jess Bridge:* A bridge for the Jess rule engine is provided in the Protégé-OWL distribution. A user interface called the SWRLJessTab is also provided to interact with this bridge. It supports the execution of SWRL rules using the Jess rule engine. The SWRLJessTab requires the Jess engine to be downloaded and installed separately as it is not open source. This engine is contained in a Java JAR called `jess.jar`, which is contained in the standard Jess distribution.
- *Fuzzy Jess Bridge:* Provides experimental support for fuzzy assertions in SWRL rules.
- *SWRL Factory:* The factory provides high-level Java APIs that support the creation and modification of SWRL rules in an OWL ontology. This API can be used to develop *linker rules* automatically inside the ArchOnt framework.

Once the bridge has been created, the inference process can be broken down into the following stages:

i. Clear all knowledge from the rule engine.

ii. Import all SWRL rules and relevant OWL knowledge from the OWL model into the bridge.
iii. Invoke the rule engine.
iv. Transfer any information asserted by a rule engine, like property values, to the OWL model.

The resulting OWL ontology, enriched with inferred knowledge, has many possible uses. For example it could be directly delivered to the end user through a compatible interface or it could be stored in a repository. In the clinical environment, these results provide means for automatically improve decision making and monitoring tasks.

The Pellet reasoner

Pellet is an OWL DL open source reasoner written in Java (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007). Like the SWRL Tab it provides an API that supports standard reasoning services, including the execution of SWRL rules[75]. Nevertheless, Pellet has recently been improved with some novel features that have a direct impact in the work with clinical archetypes and the efficiency of the integration approach.

As explained in section 4.2, the translation of archetype constraints related to quantified values is based on the OWL 2 support for embedding the definitions of user-defined data ranges in OWL ontologies. Then, a very useful capability offered by Pellet is the support for reasoning with all the built-in datatypes defined in XML Schema plus any user-defined data ranges that extend numeric or date/time derived types. Reasoning on archetypes can offer consistency checks that help validating archetypes and guaranteeing data correctness. In particular, Pellet has a datatype oracle that can check the consistency of conjunctions of (built-in or derived) XML Schema datatypes. Besides, the Pellet reasoner can detect inconsistent restrictions included in an archetype according to the RM or a parent archetype that is being specialised. The consistency checking service provided by Pellet uses the *formal definition of ontology consistency*[76] in order to ensure that an ontology does not contain any contradictory facts.

---

[75] http://clarkparsia.com/pellet/faq/rules
[76] OWL 2 Direct Semantics: http://www.w3.org/TR/owl2-semantics/#Inference_Problems

With regard to the performance of inference execution, pellet developers are working on a secondary-storage support for reasoning with large number of individuals and optimizations based on partitioning of ontologies. The performance problems mentioned in section 6.2.3 when dealing with large repositories of clinical data can be drastically reduced by means of these techniques.

7 Bindings to Clinical Ontologies

The creation of methods for properly binding archetypes and terminologies, and the capacity to do it at a semantic level, is a necessary condition to reach level 3 of SIOp. For the purpose of executing the inference, previous chapters has described an approach where archetypes are firstly translated into OWL and then enriched with SWRL rules. The translation allows for the bindings listed in the `term binding` section of the archetypes to be transformed into a set of equivalence relations between the archetype OWL classes and the SNOMED-CT OWL classes. An example of bindings attached to the *Blood Pressure* archetype is illustrated in Figure 43. SNOMED-CT codes and Physiology ontology URIs are provided for *Systolic* and *Diastolic Blood Pressures*.

An equivalence relation can be stated using the *owl:equivalentClass* built-in[77] property that links a class description **A** to another class description **B**. Then we can say that if an individual is a member of the class **A** then it also satisfies the conditions to be a member of the class **B** and viceversa. In other words, the two class descriptions involved have the same class extension or contain exactly the same set of individuals. Such symmetry is exemplified in Figure 44 by the arrows indicating two-way subsumption between *systolic* and *physiology:SystolicBloodPressure*.

---

[77] http://www.w3.org/TR/owl-ref/#equivalentClass-def

```
definition
      OBSERVATION[at0000] matches { -- Blood pressure
            data matches {
                              .
                              .
                              .
                  ELEMENT[at0004] occurrences matches {0..1} matches {  -- systolic
                              .
                              .
                              .
                  ELEMENT[at0005] occurrences matches {0..1} matches {  -- diastolic
            .
            .
            .
term_binding = <
            ["SNOMED-CT"] = <
                  items = <
                        ["at0000"] = <[SNOMED-CT(2003)::163020007]>
                        ["at0003"] = <[SNOMED-CT(2003)::364090009]>
                        ["at0004"] = <[SNOMED-CT(2003)::163030003]>
                        ["at0005"] = <[SNOMED-CT(2003)::163031004]>
                  >
            >
            ["PHYSIOLOGY ontology"] = <
                  items = <
                        ["at0004"] = <[http://www.medicalcomputing.net/owl/physiology.owl#SystolicBloodPressure]>
                        ["at0005"] = <[http://www.medicalcomputing.net/owl/physiology.owl#DiastolicBloodPressure]>
      >
```

Figure 43. Bindings examples in the *Blood Pressure* archetype.

It is worth mentioning that the establishment of such strong logical binding requires the previous finding and selection of the semantically correct external concept that will be bound to archetype components. This is difficult task that is commonly carried out by clinical experts, spending many time and effort. This issue has been addressed by the *Model Standardisation using Terminology* (MoST) methodology, and its supporting application, that provide a mechanism by which users can quickly find semantically equivalent terminology codes to bind/map to the data model fragments. Although the research focuses on the mapping of archetypes' fragments to SNOMED codes, the methodology is applicable to any other complementary model (Qamar & Rector, 2007). However, the success of this approach depends on the quality and accuracy of data mapping to terminology codes, which is still controversial as stated by Qamar, Kola, and Rector (2007).

The combination of that complementary work with the mapping techniques describe in this chapter is essential to ensure semantically equivalent coded data to achieve interoperability. This is quite helpful in practice because it encourages the following features to be implemented.

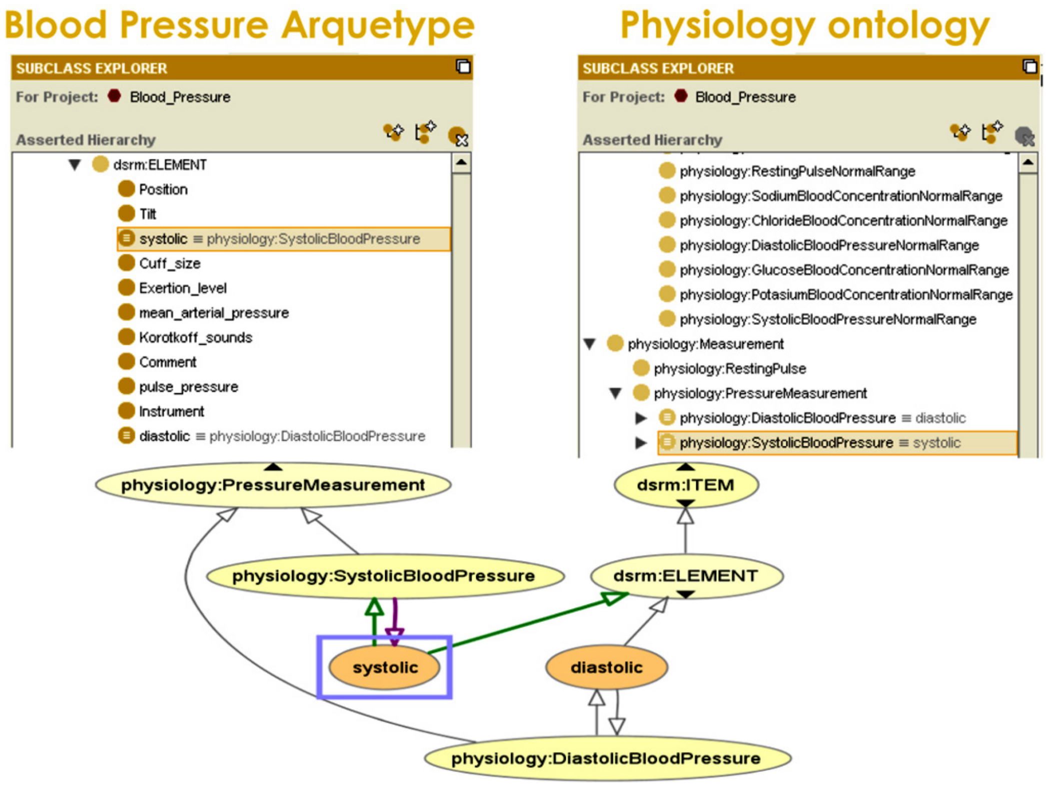

Figure 44. Protégé and OWLViz[78] view of archetypes term bindings represented by equivalence relations.

7.1 Ontology navigability

Equivalence relations allow setting bridges between both ontologies and find paths that link previously unconnected concept. For example, the OpenEHR repository organizes archetypes according to the `ENTRY` type that they specialise. That makes the *Pregnancy* archetype and the *Movement of the fetus* archetype being classified separately because the former is an `EVALUATION` while the latter is an `OBSERVATION`. Consequently the indirect path connecting both concepts through the `ENTRY` class is the only one that can be found between them.

However, a semantically more significant path is found when mapping the *Pregnancy* archetype to the SNOMED-CT concept with the same name (SNOMED code 289908002) and the *Movement of the fetus* archetype to the SNOMED-CT

[78] http://www.co-ode.org/downloads/owlviz/

Fetal movement feature (code 364617005). Then the new path can be traced through the next steps as shown in Figure 45:

i. Within the SNOMED ontology, *Fetal movement feature* is a subtype of *Fetal observable*.
ii. At the same time, the *Measure of fetus* is also related to the *Fetal observable* concept through the *is-a* SNOMED-CT attribute.
iii. Going further, this *Measure of fetus* has the *Number of fetuses* concept as one of its subtypes.
iv. Finally the SNOMED-CT *Number of fetuses* can be directly mapped or set as equivalent to the *Number of fetuses* `ELEMENT` which is part of the *Pregnancy* archetype.

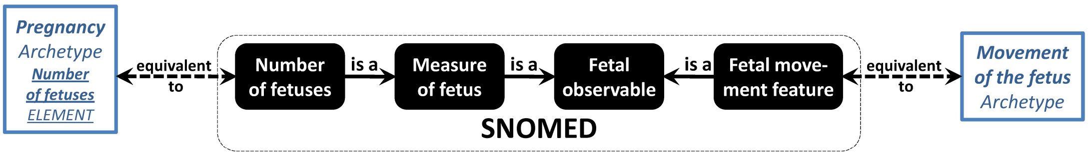

Figure 45. Archetypes connected through the SNOMED-CT ontology.

Relating archetypes from different sources supports better management and user navigation in archetype repositories. A computational technique to generate tentative archetype associations by mapping them through terms from the UMLS Metathesaurus is detailed in Lezcano, Sánchez-Alonso, and Sicilia (2010).

7.2 Checking the mapping context

The incorporation of SWRL rules into ontologies improves the reasoners' capabilities to maintain the coherence with current knowledge. Besides, the requirements for fulfilling a larger set of logical conditions, because of the equivalence relations, increase the likelihood of data correctness. For example, an archetype `ELEMENT` like *Herpes simplex* may be misused in some cases because its name accepts two different classifications.

The term "herpes simplex" may be classified as an organism (i.e. the human herpes simplex virus) or it may act as a disorder (i.e. the herpes simplex viral infection)[79]. SNOMED-CT solves this ambiguity by assigning a unique Concept ID to each one of the classifications, the *herpes simplex (organism)* is the 19965007 and the *herpes simplex (disorder)* is the 88594005. If an equivalence relation is established between the ELEMENT OWL class and the SNOMED-CT disorder concept then it has to support a number of SNOMED-CT attributes like *causative Agent*, *severity*, etc. On the other hand, if it is mapped to the organism concept, then relations through mentioned attributes are not allowed.

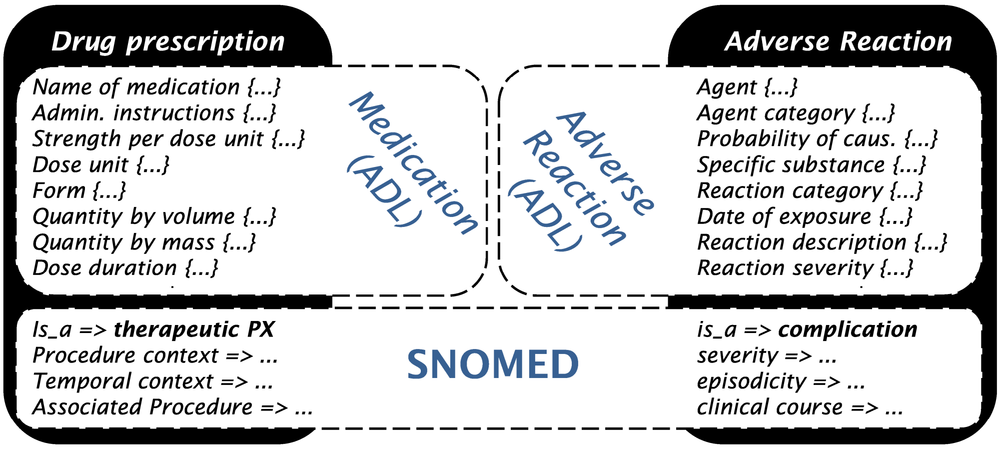

Figure 46. Archetypes and SNOMED-CT contribution in defining the *Drug Prescription* and the *Adverse Reaction* concepts.

7.3 Knowledge Integration

Figure 46 represents the combination of archetypes with the SNOMED-CT contribution in defining the *Drug Prescription* and the *Adverse Reaction* concepts after the establishment of the equivalence relations. For example, the *Adverse Reaction* concept contains the *Date of Exposure* property that comes from the archetype, while the *severity* attribute comes from SNOMED-CT. It should be noted that properties are automatically available on both sides (archetype ontology and SNOMED-CT ontology) once the equivalence is asserted. This avoids the redefinition of clinical concepts while providing a means for reusing the already existing ones.

79 In OpenGALEN, there are also two different terms *HerpesSimplexVirus* and *HerpesSimplexInfection*

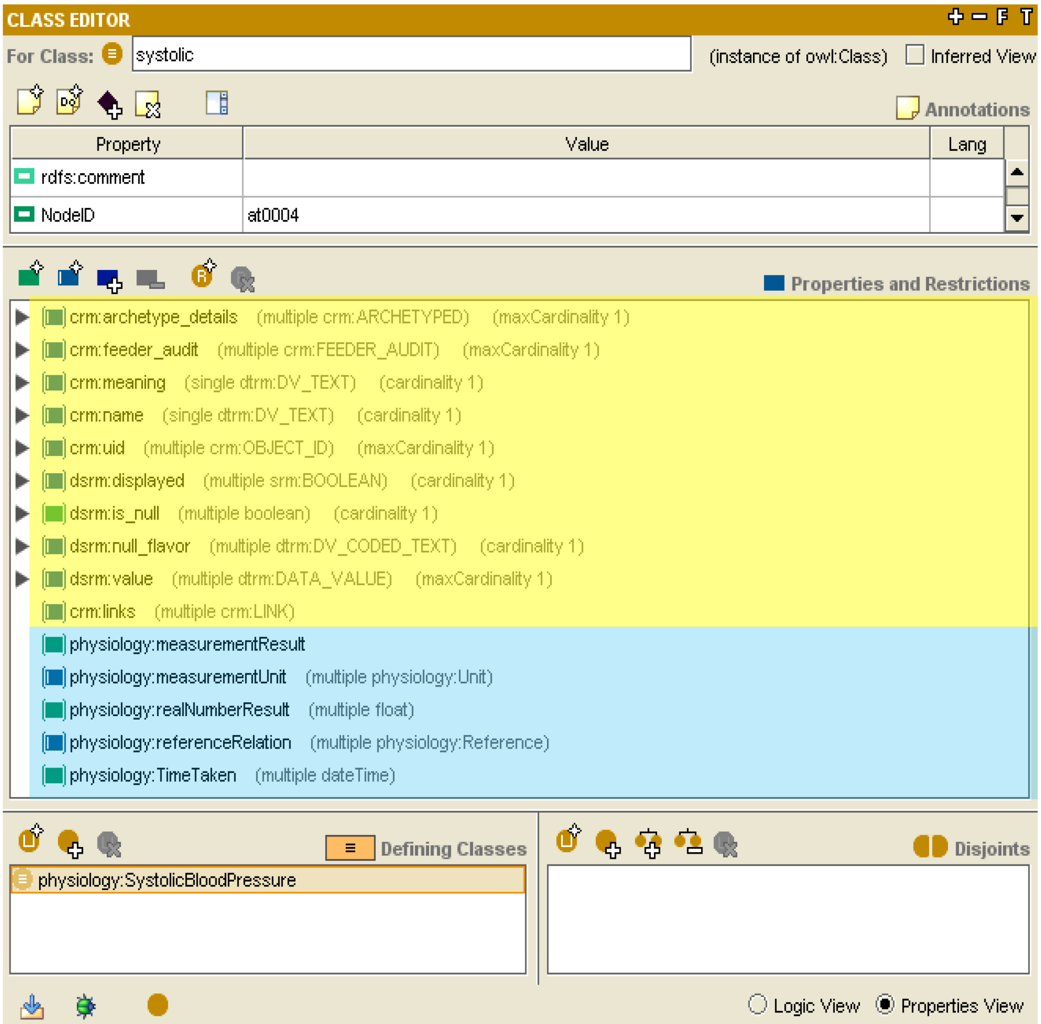

Figure 47. The screenshot shows the contribution of *Physiology* ontology and *Blood Pressure* archetype in defining the *Systolic Blood Pressure* concept. The properties in the bottom are provided by the *Physiology* ontology while the ones in the top come from the archetype.

Another example of knowledge integration can be appreciated in Figure 47, when defining the *Systolic Blood Pressure* concept with attributes coming from two different sources: a *Physiology* ontology and the openEHR *Blood Pressure* archetype.

As it was explained in the Background Chapter, the OpenEHR Model is separated in two layers: a variable one described by archetypes and a stable one described by the RM. These models are different in the levels of abstraction from models of reality such as classifications (e.g. ICD), process description (e.g. clinical guidelines), descriptive terminologies (e.g. SNOMED-CT) and ontologies like OGMS

(Ontology for General Medical Science). These models describe the real phenomena and therefore they have different types of authors, representations and purposes. However, the integration of both kinds of models is necessary to achieve SIOp and to improve patient safety.

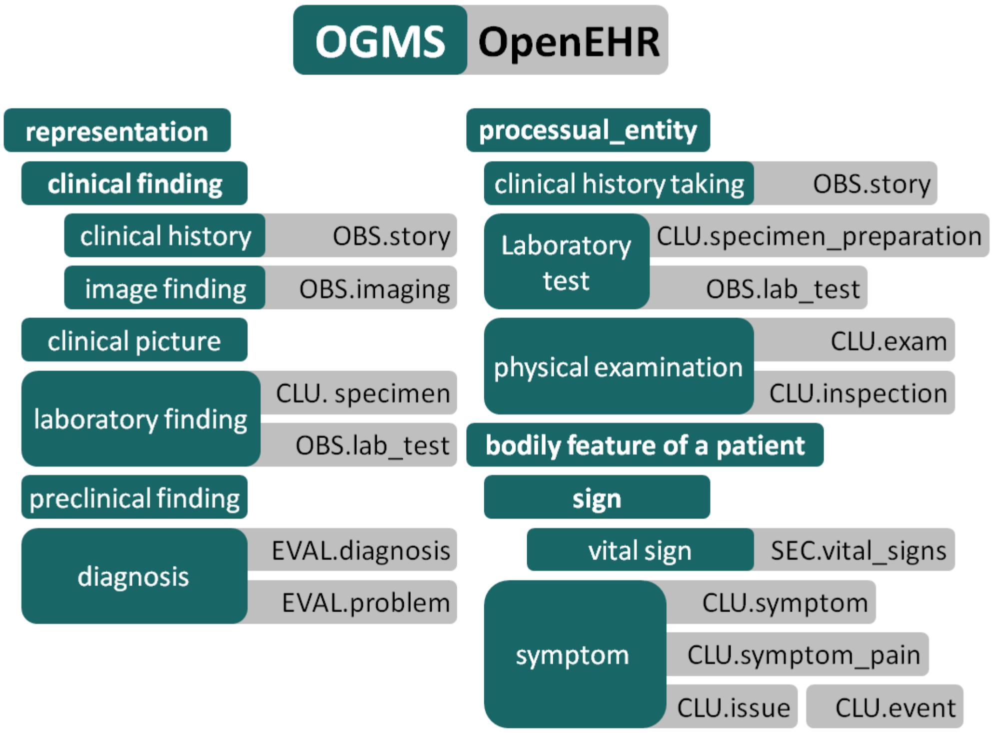

Figure 48. Tentative mapping between OGMS and openEHR archetypes.

OGMS[80,81] is a small and upper-level ontology for the domain of clinical medicine and research. Following the OBO Foundry principles, OGMS defines general terms in medicine like Disease, Disorder, Sign, Symptom, Finding, etc. It is designed to serve as anchor point for domain ontologies within medication, disease and laboratory test areas, as well as to bridge clinical medicine and basic science. The integration of OGMS and the openEHR AM has been studied as part of this thesis. Concrete benefits of such integration include the prevention of overlappings between archetype definitions as well as the reduction of ambiguities in the AM. Figure 48 illustrates a tentative mapping between OGMS concepts and clinical archetypes from the openEHR repository. Examples of incompleteness (i.e. OGMS

[80] http://code.google.com/p/ogms/

[81] The OWL version of OGMS can be downloaded from http://ogms.googlecode.com/svn/releases/2011-02-21/ontology/ogms.owl

concepts without an exact openEHR mapping) and overlapping when covering the OGMS concepts are there revealed. Then, three approaches were considered for the integration of the OGMS ontology in order to solve above mentioned problems:

i. **Direct mapping between the openEHR AM and OGMS:** As shown in Figure 48, the set of clinical archetypes currently defined in the openEHR repository is not able to completely cover the OGMS ontology.
ii. **Definition of OGMS classes as a part of the RM:** This is not very likely to happen because the openEHR philosophy is to keep the RM as stable as possible and this would clearly affect such stability.
iii. **Creation of a *Disease* layer between the RM and the AM:** This integration option is the most feasible because it preserves the RM stability while provides a common set of general concepts about the *Disease* domain to more specific clinical archetypes. It was presented by Lezcano (2009). Figure 49 shows some examples of the elements in each layer.

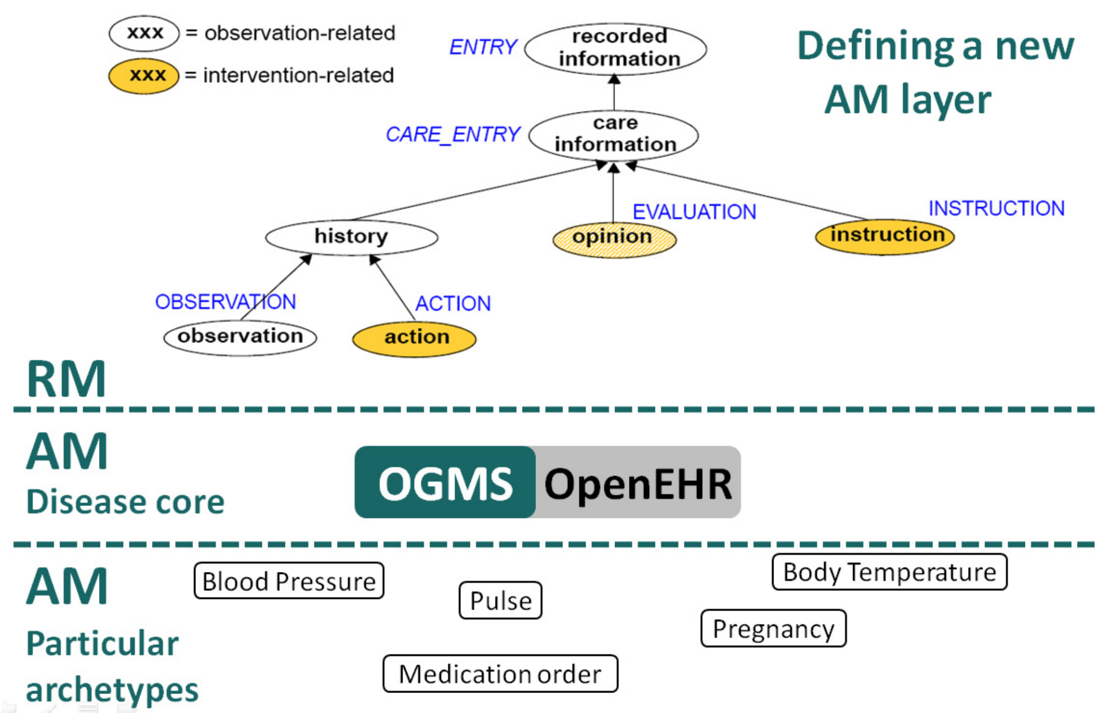

Figure 49. OGMS-openEHR integration approach.

8 Evaluation

Several case studies are presented in this chapter to show that the procedures embedded in clinical guidelines frequently contain propositions of declarative knowledge that can be expressed by a rule language like SWRL. Moreover, the feasibility of supporting clinical guidelines by the integration of archetypes, ontologies, rules and terminologies is analysed.

In addition to the *Sinuses Ligation* case study of section 6.2, the ADL2OWL translation has been employed in three clinical projects described in this chapter. According to the requirements of the those clinical projects, the implementation introduced in Chapter 5 was first oriented to the translation of `OBSERVATION` and `EVALUATION` categories of archetypes, as the *Heart rate and rhythm* archetype and the *Pulse* archetype translated in Chapter 4, and the *Intravascular Pressure* of section 6.2. Currently there are around 250 archetypes definitions in the openEHR repository and 40% of them are `OBSERVATION`s and `EVALUATION`s. After its integration as a module of the *openEHR Java Implementation Project*[82,83], the implementation is being completed in order to cover all subtypes of archetypes.

These archetypes have been authored by experts and clinicians from all over the world, moderated by the *openEHR Archetype Editorial Group*[84]. The translator has

[82] http://www.openehr.org/projects/java.html
[83] http://www.openehr.org/shared-resources/usage/academic.html
[84] http://www.openehr.org/clinicalmodels/archedgroup.html

also been tested with an archetype designed by clinicians at the Fuenlabrada Teaching Hospital[85] in Madrid (see section 8.2), as well as with three new archetypes oriented to the *Diabetic Foot* prevention and management (see section 8.3). For the automatic ADL to OWL translation to perform correct parsing and translation of any kind of archetype, the implementation must cover the entire set of RM types and all their possible combinations. In order to avoid the maintenance problems that this approach could provoke, the design process should consider as far as possible all possible constraining mechanisms defined in the AOM. This way, the ease of evolution of the implementation is fostered.

As for integration with SWRL, ten rules have been tested including the ones described in this chapter. Although the mapping mechanism has not always been the same as the one described in section 6.2.3, it should be noted that SWRL rules are defined in terms of archetype elements, thus they are totally independent of the underlying architecture. In fact, once the SWRL rules are bound to the OWL version of the archetype, their proper execution only depends on the support provided by the selected reasoner (e.g. Pellet, Jess, etc.).

This promotes the reusability and shareability of the knowledge expressed in the form of rules, which is consistent with the philosophy of open sharing of archetypes. Although rules can be executed individually, substantial results come from forward chaining reasoning and rules concatenation as the example illustrated section 8.1. With regard to the limitations of the inference process, there is no specific restriction for this OWL and SWRL integration. Thus, inference boundaries are imposed by the language itself and the reasoner (e.g. the creation of new named individuals and the use of complex mathematical functions, as fractal functions or exponential sums, which are not currently supported by SWRL, see section 2.3.2).

8.1 Decision support in the IEF-EHRS research project

The IEF-EHRS (*Information Exchange Framework for EHR Systems*) is a research project (IMADE-PIE 2009) that is been currently carried out by the

[85] http://www.madrid.org/cs/Satellite?pagename=HospitalFuenlabrada/Page/HFLA_home

Information Engineering Research Unit (*IERU*)[86] at the University of Alcalá, in combination with the *Henares Teaching Hospital*[87] and the *Alamo Consulting*[88] company. The project goals include the application of the research outcomes to clinical practice in order to improve healthcare assistance and patient security. In addition, a full CDSS[89] is being built to automatically reach to conclusions, raise alerts, as well as other monitoring aids that decrease the response time of care tasks and reduce human errors. At the same time, semantic interoperability of this framework is been taken to the highest possible level by defining a set of sharable software components that can be reused by the rest of organizations and research groups in the field. Therefore, clinical concept representation relies on clinical archetypes and the two-level approach proposed by openEHR.

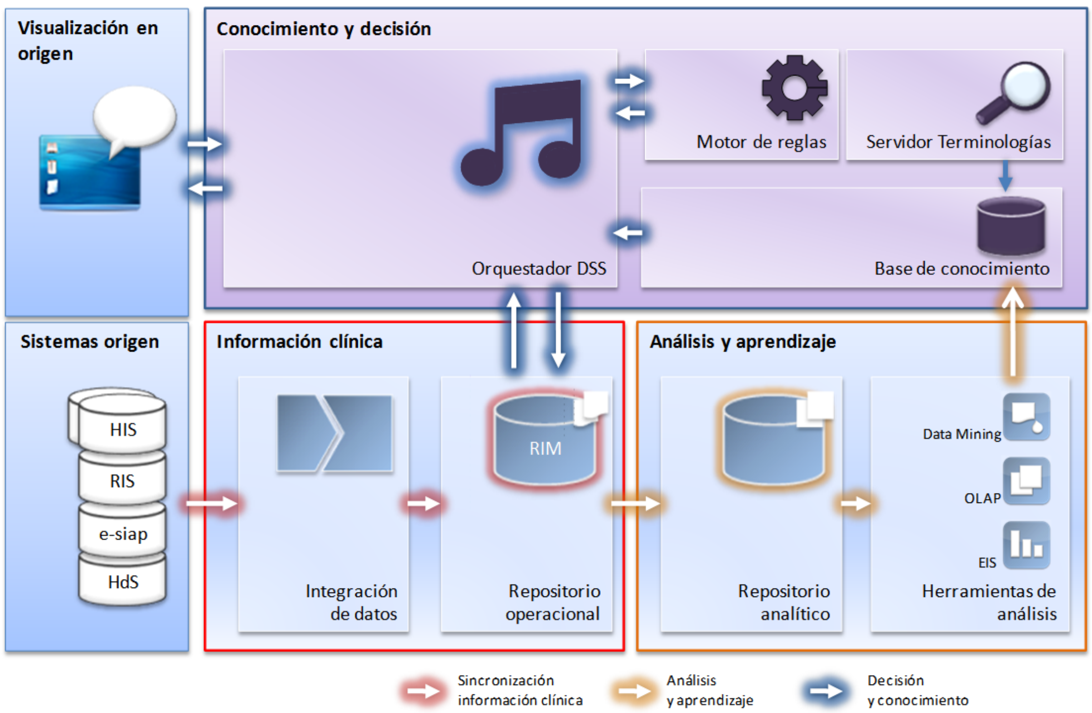

Figure 50. The IEF-EHRS project conceptual architecture.

The conceptual architecture of the IEF-EHRS project is shown in Figure 50. It includes the following main components:

---

[86] http://www.ieru.org/
[87] http://www.madrid.org/cs/Satellite?pagename=HospitalHenares/Page/HHEN_home
[88] http://www.alamoconsulting.com/
[89] Clinical Decision Support System

- **Knowledge and decision**: Represented by the violet box, it is the direct application of the translation and inference research presented in this thesis. A decision support mechanism for allergies detection that is being currently tested in the Henares Teaching Hospital is described below in this section. An “orchestrator” software inside this component sets the mappings between the patients’ data retrieved from the Clinical Information component and the required ADL definitions. Executing SWRL rules and inference is also carried out in this component.
- **Clinical Information**: Represented by the red box. It acts as a framework data cache by providing an operational repository that support querying capabilities on patients’ data. Source databases are kept synchronized with the operational repository.
- **Analysis and Learning**: Represented by the orange box. Include a set of tools that support the evolution and debugging of the clinical rules employed in the project. Statistical studies and patterns deduction is also support by Data Mining techniques applied on source patients’ data as well as on rules and guidelines implementation results.

8.1.1 The antibiotic prescribing and allergy detection case study

As pointed out in Chapter 4 of the European Commission report on Semantic Interoperability (V. N. Stroetmann et al., 2009), more complex Electronic Transfer of Prescription is considered among the areas for which reaching full SIOp or level 3 is a crucial trend. Those recommendations guided this thesis search for level 3 to the area of the new medication prescriptions that require comprehensive information on concurrent medication and details of known allergies and conditions.

Concretely, the translation of three fragments of the *Respiratory tract infections - antibiotic prescribing* published by NICE is described. The objective is to automatically recommend antibiotic prescriptions based on three SWRL rules defined according to the NICE guideline[90] and the information retrieved from the OWL version of the *Problem* archetype, downloaded from the openEHR

[90] http://guidance.nice.org.uk/CG69/

repository[91]. Results are attached to the archetype instance using the *hasInferredAlert* property that can be defined in the same OWL file that stores the SWRL rule. These results provide means for automatically improving decision making and monitoring tasks.

Immediate antibiotic prescription

The first guideline fragment states that immediate antibiotics should be offered to patients who have symptoms and signs suggestive of "Pneumonia", "Mastoiditis", "Peritonsillar abscess" or "Peritonsillar cellulitis". These disorders are uniquely identified by instances of terminology codes (e.g. the SNOMED-CT code for "Pneumonia" is 233604007). We can reuse them to fill a new class named *ImmediateAI*, that is to say *Immediate Antibiotics Infections*. Hence, the immediate prescribing depends on whether the code of the diagnosed disorder is a member of this class or not. It should be noted that further guideline modifications in the list of disorders requiring immediate prescribing will only force the update of the *ImmediateAI* class while the SWRL rule can stay unmodified.

$$
\begin{array}{rll}
Problem(?p) & \wedge & ImmediateAI(?iai) \wedge \\
hasDescribedCondition(?p, ?prob) & \wedge & value(?prob, ?cod) \wedge \\
hasConceptID(?iai, ?cod) & & \\
\Longrightarrow & & \\
hasInferredAlert(?p, \text{"306807008"}) & \wedge & \\
hasInferredAlert(?p, \text{"255631004"}) & \wedge & \\
hasInferredAlert(?p, \text{"88694003"}) & &
\end{array}
$$

SNOMED terms involved:
255631004 - Antibiotic
88694003 - Immediate
306807008 - Recommendation to start drug treatment

Figure 51. Antibiotic prescribing - SWRL rule (1st fragment).

The diagnostic data is collected from the *Problem* archetype that is designed to record a condition or issue defined by a clinician who is deemed summative of a range of symptoms of the person. The archetype provides the SNOMED-CT codes for the *described condition* and for the *attributed diseases*. If any of them is a member of the *ImmediateAI* class, then the patient is considered to be at risk of developing complications so the rule raises an alert that recommends immediate antibiotic prescribing. Such rule should include a disjunction operator as there are two `ELEMENT`s to check in the archetype, and the same happens in the other two guideline fragments that are described below. However, SWRL does not support

[91] http://openehr.org/knowledge/

disjunctions of atoms, so they are checked in separated rules[92]. Figure 51 shows the rule that queries the *described condition* according to the guideline's first fragment.

$$\begin{array}{rcl} hasAgeOnsetELT(?p, ?a) & \wedge & Problem(?p) \wedge \\ hasDescribedCondition(?p, ?prob) & \wedge & value(?a, ?age) \wedge \\ swrlb: lessThanOrEqual(?age, 2) & \wedge & value(?prob, \text{“194290005”}) \\ \Longrightarrow & & \\ hasInferredAlert(?p, \text{“306807008”}) & \wedge & \\ hasInferredAlert(?p, \text{“255631004”}) & \wedge & \\ hasInferredAlert(?p, \text{“88694003”}) & & \end{array}$$

SNOMED terms involved:
255631004 - Antibiotic
88694003 - Immediate
306807008 - Recommendation to start drug treatment
194290005 - Bilateral acute otitis media

Figure 52. Antibiotic prescribing - SWRL rule (2nd fragment).

As explained in section 6.2.2, the *hasDescribedCondition* property is previously filled by a *linker rule* in order to rapidly access the *described condition* value in further clinical rules (i.e., first, second and third SWRL fragment representation of the guideline). Linker rules are also used to fill the *hasAgeOnsetELT* property, which is then used in the second fragment representation and with the *hasMedicationSubstance* and *hasReactionSubstance* properties from the allergy detection SWRL rule shown in Figure 54.

$$\begin{array}{rcl} Problem(?p) & \wedge & DelayedAI(?dai) \wedge \\ hasDescribedCondition(?p, ?prob) & \wedge & value(?prob, ?cod) \wedge \\ hasConceptID(?dai, ?cod) & & \\ \Longrightarrow & & \\ hasInferredAlert(?p, \text{“306807008”}) & \wedge & \\ hasInferredAlert(?p, \text{“255631004”}) & \wedge & \\ hasInferredAlert(?p, \text{“75784006”}) & & \end{array}$$

SNOMED terms involved:
255631004 - Antibiotic
75784006 - Delayed
306807008 - Recommendation to start drug treatment

Figure 53. Antibiotic prescribing - SWRL rule (3rd fragment).

Depending on the clinical assessment of severity, the second fragment analyzed here states that an immediate prescribing strategy should be agreed for children younger than 2 years with “Bilateral acute otitis media”. There is a single disorder to check in this case so it makes no sense to define a new class; the SNOMED-CT code can be directly used. The patient's age is obtained from the *Age at initial onset* `ELEMENT`, also included in the OWL version of the *Problem* archetype. The antecedent part of the SWRL rule in Figure 52 considers all these parameters, including the age comparison, while the consequent part is the same as the former rule because the prescribing strategy is also immediate.

---

[92] The *SWRL-FOL* extends SWRL by adding the standard logical connectives such as negation and disjunction from first order logic in spite of the fact that their addition may complicate the language semantics, http://www.w3.org/Submission/2005/SUBM-SWRL-FOL-20050411/

Delayed antibiotic prescribing

The third guideline fragment states that a delayed antibiotic prescribing strategy should be considered for patients with “Acute otitis media”, “Acute pharyngitis/Acute tonsillitis”, “Common cold”, “Acute rhinosinusitis” or “Acute bronchitis”. A new class named *DelayedAI* or *Delayed Antibiotics Infections* is defined in an analogous manner with the first fragment translation. Figure 53 illustrates the new rule that attaches SNOMED-CT codes recommending delayed prescribing when one of the above disorders (classified as *DelayedAI*) is detected. The consequent part is also similar to the one in the first rule. In contrast the priority qualifier has been set to “Delayed”.

Validating medical prescriptions

Working in the context of abnormal reactions and allergies to medications, where decision support can substantially improve support to the medication process, the OWL version of two archetypes (*Medication* and *Adverse Reaction*) and a subset of SNOMED-CT are taken as the point of departure (see section 2.4). The objective is to prevent interactions according to known allergies stored in the patient’s EHR.

$$
\begin{aligned}
&Medication(?med) \wedge AdverseReaction(?ar) \wedge \\
&hasMedicationSubstance(?med, ?x) \wedge DrugAllergy(?dga) \wedge \\
&hasReactionSubstance(?ar, ?y) \wedge causativeAgent(?dga, ?x) \wedge \\
&causativeAgent(?dga, ?y) \\
&\Longrightarrow \\
&hasInferredAlert(?med, \text{“281647001”})
\end{aligned}
$$

SNOMED terms involved:
281647001 - Adverse Reaction

Figure 54. Allergy detection - SWRL rule.

The *Medication* archetype specifies the description of the medication as part of an `INSTRUCTION` or `ACTION` record taken with respect to medication. This will usually occur in response to a medication order or prescription, but may be self administered or supplied by a pharmacy. The archetype provides the SNOMED-CT code that links with the term that represents the *medication substance*. It also includes a *Deferred supply* `ELEMENT` that is set if the medication supply is delayed, as occurs in the previous mentioned guideline.

In other direction, the *Adverse Reaction* archetype is usually employed for recording anomalous reaction(s) to a particular 'Agent'. The SNOMED-CT code corresponding to the *reaction substance* is provided. Thus, it may be specified that the decision on whether the prescription should be approved or not depends on the *proximity* between both terms, the *medication substance* and the *reaction substance*, inside the SNOMED-CT ontology. Such *proximity* can be traced by following the *causative Agent* relationship that links every allergy with the drugs or substances that caused it. For example, the *Co-fluampicil allergy* is related to *Ampicillin* and *Floxacillin* through the *causative Agent* relationship. Therefore an *Ampicillin* prescription must not be issued to a patient whose EHR stores a sample of the *Adverse Reaction* archetype containing the code for *Floxacillin* and viceversa. Once the problem is detected, a new alert is attached to the instance of *Medication* archetype using the *hasInferredAlert* property as explained previously in this section.

In the SNOMED-CT ontology the *causative Agent* is a relationship between the *DrugAllergy* class and the *Drug* class. Thus the SWRL rule in Figure 54 captures the allergy detection algorithm described above. This rule can be invoked independently as an aid to the prescription validation process. Nevertheless, the benefits are more evident when SWRL rules are chained than when they are isolated. In fact, the execution of the antibiotic prescribing rules in this section can be considered as a triggering event for the rule in Figure 54.

The flow chart in Figure 55 describes the control flow for the combined execution of these rules. It should be noted that the process cannot be carried out without human intervention in between the inference execution blocks. For example, a doctor must select the type of antibiotics and the active ingredient that is going to be offered to the patient between the antibiotic prescribing alert, which is raised in the left side of Figure 55, and the moment when the prescription becomes effective in the right side. Still, it is a reliable method that improves patient security during diagnosis and treatment. Also, execution scalability is well supported as newly prescribing guidelines can be concurrently launched in the left side, previously to the prescribing validation stage in the right side.

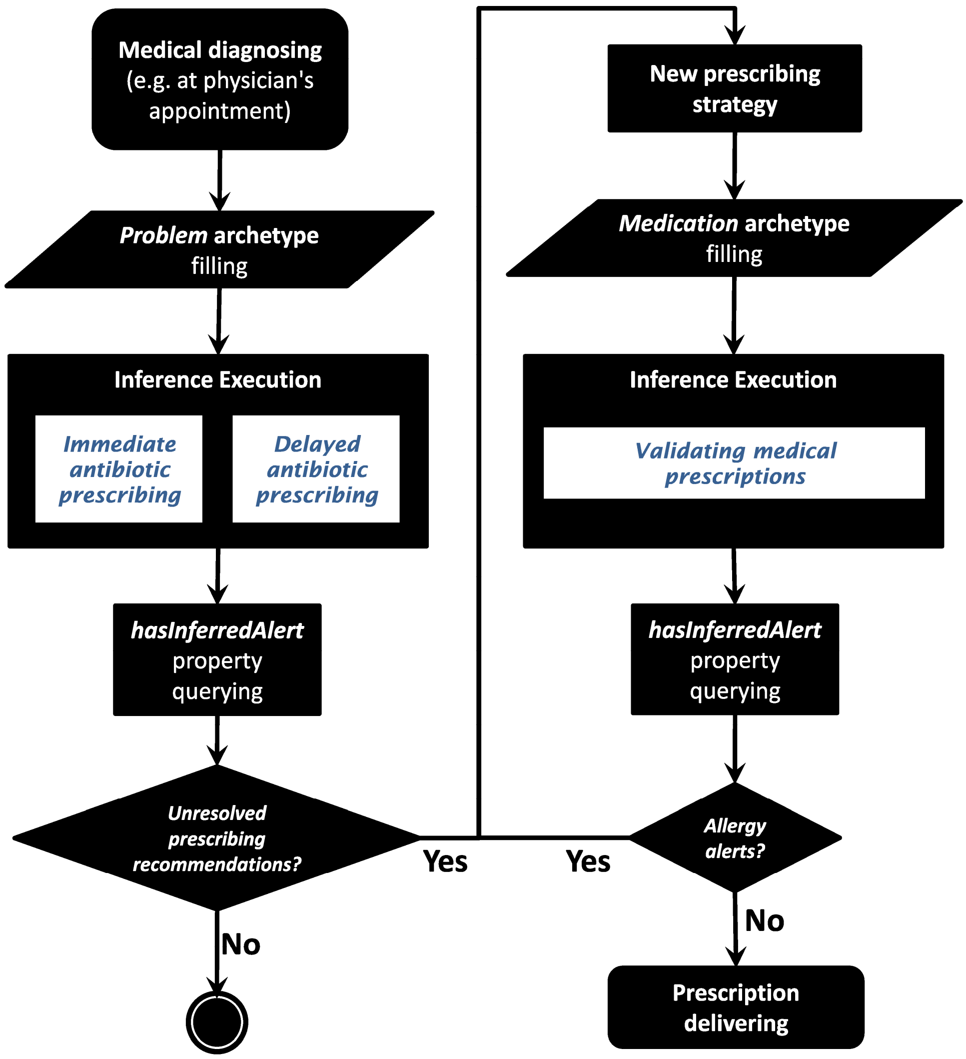

Figure 55. Control flow for the chained execution of SWRL rules as an aid to decision making in antibiotic prescribing and allergy detection.

8.2 Alerts in the CISEP project

The CISEP (*Intelligent Clinical Records for Patient Safety*) is a finished research project (code FIT-350301-2007-18), that was funded by the Spanish Ministry of Science and Technology. The project was carried out by the *Fuenlabrada Teaching Hospital*[93], the *iSOCO*[94] company and several universities, including the *University*

93 http://www.madrid.org/cs/Satellite?pagename=HospitalFuenlabrada/Page/HFLA_home

94 iSOCO was founded in 1999 as a spin-off by the Spanish National Research Council, with the idea to put academic investigation on the market. http://www.isoco.com/

of Carlos III, the *European University of Madrid* and the *University of Alcalá*, specifically the Information Engineering Research Unit (*IERU*).

The main objective of the project was to design, develop and implement a healthcare system based on information processing techniques and semantic modelling. Such system was oriented to offer value added services working from the electronic health records, in order to improve patient safety. A key goal in this project was to guarantee the interoperability of the heterogeneous software components that were functioning in the hospital.

Concrete objectives included supporting alerts and decision making aids within a specific subset of patient security problems like medication, primary and secondary care. Pressure ulcer management is one of those security problems for which monitoring aids are essential in the *Fuenlabrada Hospital*. Following the approach presented in this thesis a pilot service was developed and tested for pressure ulcer prevention. Section 8.2.1 describes the archetype and clinical rules integration that was undertaken in such pilot service.

8.2.1 The pilot service for pressure ulcer prevention

A clinical guideline for pressure ulcer risk assessment and prevention has been published by NICE[95]. According to the guideline, treatment and care should take into account patients' individual needs and preferences. In addition, a good communication is essential to properly asses several risk factors. The *Fuenlabrada Hospital* has defined the *Norton* archetype as an `OBSERVATION`, in order to keep track of **five risk factors** that must be considered when assessing the pressure ulcer risk. The risk factors are: *Activity*, *Incontinence*, *Mobility*, *Mental Condition* and *Physical Condition*. A level of risk (*integer*) is assigned to each one of them in a way that the higher the value, the better the status. The ADL definition is shown in Figure 56. It was translated to OWL according to the translation principles described in Chapter 4 and the implementation in Chapter 5, in order to attach it SWRL rules. In this manner, a significant part of the evaluation of the data contained in the *Norton* archetype is automatically inferred. Satisfactory results were gathered from the first tests, so subsequent goals included the adjustment of the inference

[95] http://guidance.nice.org.uk/CG29/

execution frequencies in order to maximize patients' safety within an affordable processing power.

```
archetype (adl_version=1.4)
  openEHR-EHR-OBSERVATION.Norton.v1draft

concept
  [at0000]  -- Escala de Norton
language
  original_language = <[ISO_639-1::es]>
  translations = <
    ["en"] = <
      language = <[ISO_639-1::en]>
      author = <
        ["name"] = <"????">
      >
    >
  >
description
  original_author = <
    ["name"] = <"Pablo Serrano">
  >
  details = <
    ["es"] = <
      language = <[ISO_639-1::es]>
      purpose = <"">
      use = <"">
      misuse = <"">
    >
    ["en"] = <
      language = <[ISO_639-1::en]>
      purpose = <"*(es)">
      use = <"*(es)">
      misuse = <"*(es)">
    >
  >
  lifecycle_state = <"Initial">
  other_contributors = <>
  other_details = <
    ["references"] = <"">
  >

definition
  OBSERVATION[at0000] matches {  -- Escala de Norton
    data matches {
      HISTORY[at0001] matches {  -- Event Series
        events cardinality matches {1..*; unordered} matches {
          EVENT[at0002] occurrences matches {0..*} matches {  -- Cualquier evento
            data matches {
              ITEM_LIST[at0004] matches {  -- List
                items cardinality matches {0..*; unordered} matches {
                  ELEMENT[at0005] occurrences matches {0..1} matches {--Estado físico general
                    value matches {
                      1|[local::at0010],   -- Muy malo
                      2|[local::at0009],   -- Regular
                      3|[local::at0008],   -- Mediano
                      4|[local::at0006]    -- Bueno
                    }
                  }
                  ELEMENT[at0007] occurrences matches {0..1} matches {  -- Estado mental
                    value matches {
                      1|[local::at0011],   -- Estuporoso
                      2|[local::at0012],   -- Confuso
                      3|[local::at0013],   -- Apático
                      4|[local::at0014]    -- Alerta
                    }
                  }
                  ELEMENT[at0015] occurrences matches {0..1} matches {  -- Actividad
                    value matches {
                      1|[local::at0018],   -- Encamado
                      2|[local::at0019],   -- Sentado
                      3|[local::at0020],   -- Camina con ayuda
                      4|[local::at0021]    -- Ambulante
                    }
                  }
                  ELEMENT[at0016] occurrences matches {0..1} matches {  -- Movilidad
                    value matches {
                      1|[local::at0022],   -- Inmóvil
                      2|[local::at0023],   -- Muy limitada
```

```
                        3|[local::at0024],   -- Disminuida
                        4|[local::at0025]    -- Total
                      }
                    }
                    ELEMENT[at0017] occurrences matches {0..1} matches {  -- Incontinencia
                      value matches {
                        1|[local::at0026],   -- Urinaria y fecal
                        2|[local::at0027],   -- Urinaria
                        3|[local::at0028],   -- Ocasional
                        4|[local::at0029]    -- Ninguna
                      }
                    }
                    ELEMENT[at0031] occurrences matches {0..1} matches {  -- Puntos totales
                      value matches {
                        DV_COUNT matches {
                          magnitude matches {|5..20|}
                        }
                      }
                    }
                  }
                }
              }
            }
            EVENT[at0032] occurrences matches {0..1} matches {  -- 6 horas post ingreso
              data matches {
                use_node ITEM_LIST /data[at0001]/events[at0002]/data[at0004]
              }
            }
            EVENT[at0033] occurrences matches {0..1} matches {  -- cambio de estado
              data matches {
                use_node ITEM_LIST /data[at0001]/events[at0002]/data[at0004]
              }
            }
          }
        }
      }
    }

ontology
    terminologies_available = <"SNOMED-CT", ...>
    term_definitions = <
      ["es"] = <
        items = <
          ["at0000"] = <
            text = <"Escala de Norton">
            description = <"Escala de evaluación del riesgo de lesiones por presión">
          >
          ["at0001"] = <
            text = <"Event Series">
            description = <"@ internal @">
          >
          ["at0002"] = <
            text = <"Cualquier evento">
            description = <"*">
          >
          ["at0004"] = <
            text = <"List">
            description = <"@ internal @">
          >
          ["at0005"] = <
            text = <"Estado físico general">
            description = <"*">
          >
          ["at0006"] = <
            text = <"Bueno">
            description = <"Bueno">
          >
          ["at0007"] = <
            text = <"Estado mental">
            description = <"*">
          >
          ["at0008"] = <
            text = <"Mediano">
            description = <"Mediano">
          >
          ["at0009"] = <
            text = <"Regular">
            description = <"Regular">
          >
          ["at0010"] = <
            text = <"Muy malo">
            description = <"Muy malo">
```

```
            >
            ["at0011"] = <
              text = <"Estuporoso">
              description = <"Estuporoso">
            >
            ["at0012"] = <
              text = <"Confuso">
              description = <"Confuso">
            >
            ["at0013"] = <
              text = <"Apático">
              description = <"Apático">
            >
            ["at0014"] = <
              text = <"Alerta">
              description = <"Alerta">
            >
            ["at0015"] = <
              text = <"Actividad">
              description = <"*">
            >
            ["at0016"] = <
              text = <"Movilidad">
              description = <"*">
            >
            ["at0017"] = <
              text = <"Incontinencia">
              description = <"*">
            >
            ["at0018"] = <
              text = <"Encamado">
              description = <"Encamado (precisa cambios posturales)">
            >
            ["at0019"] = <
              text = <"Sentado">
              description = <"Ayuda de 2 ó más personas para levantarse o deambular, sentado">
            >
            ["at0020"] = <
              text = <"Camina con ayuda">
              description = <"Ayuda de una persona para levantarse o deambular">
            >
            ["at0021"] = <
              text = <"Ambulante">
              description = <"Autonomía">
            >
            ["at0022"] = <
              text = <"Inmóvil">
              description = <"*">
            >
            ["at0023"] = <
              text = <"Muy limitada">
              description = <"*">
            >
            ["at0024"] = <
              text = <"Disminuida">
              description = <"*">
            >
            ["at0025"] = <
              text = <"Total">
              description = <"*">
            >
            ["at0026"] = <
              text = <"Urinaria y fecal">
              description = <"*">
            >
            ["at0027"] = <
              text = <"Urinaria">
              description = <"*">
            >
            ["at0028"] = <
              text = <"Ocasional">
              description = <"*">
            >
            ["at0029"] = <
              text = <"Ninguna">
              description = <"*">
            >
            ["at0031"] = <
              text = <"Puntos totales">
              description = <"Riesgo muy alto para puntuaciones entre 5 y 11, alto para
                            puntuaciones entre 12 y 14, sin riesgo para puntuaciones entre 15 y
                            20">
```

```
          >
          ["at0032"] = <
            text = <"6 horas post ingreso">
            description = <"*">
          >
          ["at0033"] = <
            text = <"cambio de estado">
            description = <"*">
          >
        >
      >
```

Figure 56. The OBSERVATION.*Norton* archetype as defined by the *Fuenlabrada Hospital.*

The *Total Points* ELEMENT (at0031) is a DV_COUNT designed to store the sum of the five risk factors values. The risk assessment rules are based on the value of this ELEMENT. If the archetype instance is completed without filling the *Total Points* ELEMENT, then the linker rule shown in Figure 57 can fill it. Although some approaches could give more weights to some risk factors than others, in the implemented pilot service the weight distribution was homogeneous.

$$
\begin{array}{rcl}
Patient(?pat) & \wedge & hasActivity(?pat,?act) \wedge \\
hasIncontinence(?pat,?inc) & \wedge & hasMobility(?pat,?mob) \wedge \\
hasMentalCond(?pat,?men) & \wedge & hasPhysicalCond(?pat,?phy) \wedge \\
add(?subt1,?act,?inc) & \wedge & add(?subt2,?mob,?men) \wedge \\
add(?subt3,?subt1,?subt2) & \wedge & add(?total,?subt3,?phy) \\
\Longrightarrow & & \\
 & & hasTotalNortonPoints(?pat,?total)
\end{array}
$$

Figure 57. Linker rule to fill the *Total Points* ELEMENT.

It should be noted that the linker rule in Figure 57 depends in turn on several linker rules that fill each risk factor property (i.e., *hasActivity*, *hasMobility*, etc.). Then three clinical rules are defined to make the automatic assessment of pressure ulcer risk. Allowed values for the risk are: *"none"*, *"high"*, *"very high"*. The SWRL rules representing the guideline are shown in Figure 58. The resulting OWL ontology, enriched with inferred knowledge, provide means for automatically improve decision making and monitoring tasks on the pressure ulcer management field.

$$
\begin{aligned}
Patient(?pat) &\wedge hasTotalNortonPoints(?pat, ?tot) \wedge \\
greaterThan(?tot, 11) &\wedge lessThan(?tot, 15) \\
\Longrightarrow & \\
& hasNortonRisk(?pat, \text{“high”})
\end{aligned}
$$

$$
\begin{aligned}
Patient(?pat) &\wedge hasTotalNortonPoints(?pat, ?tot) \wedge \\
greaterThan(?tot, 4) &\wedge lessThan(?tot, 12) \\
\Longrightarrow & \\
& hasNortonRisk(?pat, \text{“very high”})
\end{aligned}
$$

$$
\begin{aligned}
Patient(?pat) &\wedge hasTotalNortonPoints(?pat, ?tot) \wedge \\
greaterThan(?tot, 14) & \\
\Longrightarrow & \\
& hasNortonRisk(?pat, \text{“none”})
\end{aligned}
$$

Figure 58. SWRL clinical rules for pressure ulcer risk assessment.

8.3 Other current projects and directions

The experiences earned in the previously sketched projects are being applied in the new contexts described in this section, serving as a complement to the Future Work Chapter.

8.3.1 The Clinical DSS research and development project

The Clinical DSS (Clinical and Social Security Decisions Support System) is a research and development project currently carried out by a consortium of private enterprises and public universities and research organisations. The main objective is to develop a set of applications to support, in first place, the integration and homogenization of the clinical data retrieved from several and heterogeneous healthcare systems. Then the set of applications must be able to react to certain clinical episodes by triggering alerts and/or making recommendations in a way that resembles a doctor's procedure when acceding and assessing current and historical EHRs. It has been structured as a cooperation project, in order to take advantage of the particular technological capabilities offered by each part. Partners' names are listed below:

- Enterprises:
  - *Álamo Consulting*
  - *iSoft*[96]
  - *Oracle*[97]
  - *BITAC*[98]
- Universities
  - *University of Alcalá*, specifically the Information Engineering Research Unit (*IERU*).
  - *Polytechnic University of Valencia*[99]
- Research organizations:
  - *TicSalut*[100]
  - *Healthcare Technology Evaluation Agency*[101]
  - *Iavante*[102]
  - *Fuenlabrada Teaching Hospital*

Given that many and heterogeneous technologies and systems will meet in unique context, the highest levels of semantic interoperability (section 2.1) are required to guarantee a fast and proper development of the project. In the context of knowledge representation and inference, such interoperability is being encouraged by the *IERU* by applying the *archetypes-rules* integration approach studied in this thesis. Thus, the IERU concrete objectives within the Clinical DSS project are:

- Representation of the clinical concepts, parameters and measures involved in the risk assessment and prevention of the *Diabetic Foot* (SNOMED code 280137006) by means of a constraint model that guarantees interoperability, that is to say, openEHR archetypes. Further clinical cases that will be under study are *Colon Cancer* and *Specialized Care*. The following archetypes have been defined by the *IERU* with the medical advice provided by *TicSalut* experts. These archetypes contain

---

96 http://www.isoftsanidad.es/
97 http://www.oracle.com/
98 http://www.bitac.com/
99 http://www.upv.es/
100 http://www.gencat.cat/salut/ticsalut/
101 http://www.isciii.es/htdocs/en/investigacion/Agencia_quees.jsp
102 http://www.iavante.es/

the bindings to SNOMED codes in order to support the implementation of the benefits described in Chapter 7.

- o ***openEHR-EHR-OBSERVATION.inspection-nails***: For recording the findings on inspection of toenails like deformity, incorrect cut, etc.
- o ***openEHR-EHR-OBSERVATION.hyperglycemia_test***: For recording the results from *Chronic Hyperglycemia* tests as the *HbA1c* and the *Glucose Level*.
- o ***openEHR-EHR-EVALUATION.diabetes***: An evaluative statement for the conclusions by a clinician about Diabetes presence.

- ADL to OWL translation of the above mentioned archetypes, according to the translation principles and implementation described in chapters 4 and 5.
- Representation of the clinical rules for *Diabetic Foot* prevention through a semantic web language as SWRL. These rules will be previously defined on the basis of the tacit knowledge of medical experts and the results from the automatic learning and data mining techniques provided by the *Polytechnic University of Valencia*. This and subsequent objectives are still under development.
- Instantiation of the archetypes OWL version to capture the significant patient clinical data retrieved from several biomedical sources. The semantic capabilities of Oracle Database Semantic Technologies[103] are being tested in order to increase the implementation performance.
- Inference execution. The selection of the semantic reasoner to use from the currently available products is still ongoing.

[103] As part of Oracle Spatial 11g, an option for Oracle Database 11g Enterprise Edition, Oracle delivers an advanced semantic data management capability not found in any other commercial or open source triple store. With native support for RDF/RDFS/OWL/SKOS standards, this semantic data store enables application developers to benefit from an open, scalable, secure, integrated, efficient platform for RDF and OWL-based applications. These semantic database features enable storing, loading, and DML access to RDF/OWL data and ontologies, inference using RDFS, OWL and SKOS semantics and user-defined rules, querying of RDF/OWL data and ontologies using SPARQL-like graph patterns embedded in SQL, and ontology assisted querying of enterprise (relational) data.

8.3.2 Diagnosing the Obstructive Chronic Pulmonary Disease

The *IERU* at the University of Alcalá has recently begun to study the representation and automatic inference opportunities when diagnosing the Obstructive Pulmonary Disease (COPD)[104].

$$
\begin{aligned}
Patient(?pat) \quad & \wedge \quad hasFEV1\_FVC\_DVQuantity(?pat, ?qratv) \wedge \\
greaterThanOrEqual(?qratv, 70) \quad & \\
\Longrightarrow \quad & \\
& hasStageCOPD(?pat, \text{“}none\text{”})
\end{aligned}
$$

$$
\begin{aligned}
Patient(?pat) \quad & \wedge \quad hasFEV1\_FVC\_DVQuantity(?pat, ?qratv) \wedge \\
hasFEV1\_DVQuantity(?pat, ?gfevv) \quad & \wedge \quad lessThan(?qratv, 70) \wedge \\
greaterThanOrEqual(?gfevv, 80)) \quad & \\
\Longrightarrow \quad & \\
& hasStageCOPD(?pat, \text{“}mild\text{”})
\end{aligned}
$$

$$
\begin{aligned}
Patient(?pat) \quad & \wedge \quad hasFEV1\_FVC\_DVQuantity(?pat, ?qratv) \wedge \\
hasFEV1\_DVQuantity(?pat, ?gfevv) \quad & \wedge \quad lessThan(?qratv, 70) \wedge \\
greaterThanOrEqual(?gfevv, 50)) \quad & \wedge \quad lessThan(?gfevv, 80) \\
\Longrightarrow \quad & \\
& hasStageCOPD(?pat, \text{“}moderate\text{”})
\end{aligned}
$$

$$
\begin{aligned}
Patient(?pat) \quad & \wedge \quad hasFEV1\_FVC\_DVQuantity(?pat, ?qratv) \wedge \\
hasFEV1\_DVQuantity(?pat, ?gfevv) \quad & \wedge \quad lessThan(?qratv, 70) \wedge \\
greaterThanOrEqual(?gfevv, 30)) \quad & \wedge \quad lessThan(?gfevv, 50) \\
\Longrightarrow \quad & \\
& hasStageCOPD(?pat, \text{“}severe\text{”})
\end{aligned}
$$

$$
\begin{aligned}
Patient(?pat) \quad & \wedge \quad hasFEV1\_FVC\_DVQuantity(?pat, ?qratv) \wedge \\
hasFEV1\_DVQuantity(?pat, ?gfevv) \quad & \wedge \quad lessThan(?qratv, 70) \wedge \\
lessThan(?gfevv, 30) \quad & \\
\Longrightarrow \quad & \\
& hasStageCOPD(?pat, \text{“}very\_severe\text{”})
\end{aligned}
$$

Figure 59. SWRL rules to infer the stage of COPD according to the GOLD guidelines.

According to the clinical guideline for *Diagnosis, Management, and Prevention of COPD* published by GOLD[105,106], a diagnosis of COPD should be considered if any of the following indicators are present in an individual over age 40: *Dyspnea*, *Chronic Cough*, *Chronic Sputum Production* and/or history of exposure to risk

---

[104] COPD is a preventable and treatable disease with some significant extrapulmonary effects that may contribute to the severity in individual patients. Its pulmonary component is characterized by airflow limitation that is no fully reversible. The airflow limitation is usually progressive and associated with an abnormal inflammatory response of the lung to noxious particles of gases.

[105] GOLD - Global Initiative for Chronic Obstructive Lung Disease

[106] http://www.goldcopd.com/guidelinesresources.asp

factors (especially cigarette smoking). However, there is incomplete evidence that the individuals who meet these indicators necessarily progress on to develop COPD. Therefore, the diagnosis should be confirmed by a Spirometry test[107] that allows classifying the severity of COPD into four stages: *Stage I - Mild COPD*, *Stage II – Moderate COPD, Stage III – Severe COPD* and *Stage IV – Very Severe*. At this point, inference execution can automatically deduce such COPD stage in order to trigger the search for underlying causes and to recommend a therapy procedure according to the severity of the stage.

As in previous decision support examples, one or more archetypes are required to homogeneously represent the patient's data that will be used during inference. In this case, the openEHR archetype repository does not provide a Spirometry archetype or any other definition supporting the representation of Spirometry results. Therefore, an `OBSERVATION`.*spirometry* was created, in collaboration with medical experts at *Henares Teaching Hospital*. The archetype includes the *Forced Expiratory Volume in 1 Second* (FEV_1) and the ratio of FEV_1 to the *Forced Vital Capacity* (FVC), which will be used in the clinical rules in Figure 59 to infer the stage of COPD and attached it to the patient OWL instance.

It should be noted that a Stage 0 or "none" has been added to the four stages defined by GOLD, allowing to fill the *hasStageCOPD* property of every patient, regardless of being healthy or not. It should also be noted that evaluated ranges of FEV_1 have an empty intersection, so any patient can receive two different assessments at the same time. Once the stage is inferred, the appropriate therapy can be followed from the GOLD guideline.

[107] The Spirometry test measures the amount of air a person can breathe out, and the amount of time taken to do so.

9 Conclusions

Different approaches to support the interoperability between heterogeneous systems are currently being conceived by enterprises and researchers in the Healthcare domain. Along with that, large clinical terminologies like SNOMED-CT and information models like the one proposed by the openEHR Foundation are developed. The objective of present thesis, as stated in the introductory chapter, was to extend such previous efforts and to enhance the archetype approach to clinical information and knowledge representation by means of ontologies and semantic web languages, in order to support features that correspond to level 3 of SIOp between EHRs. This main objective has been successfully achieved from several angles and perspectives, defined by each one of the concrete objectives.

It has been shown that it is possible to integrate patients' data, clinical knowledge and terminologies in a way that is meaningful to computers, allowing to automatically interpret the evidence and making possible to reach significant clinical conclusions. When provided to healthcare systems, such conclusions are used to trigger alerts as well as to generate monitoring recommendations and decision making aids that considerably improve patient safety.

Concretely, archetypes have been extended with computational semantics by translating them into OWL. A translation method was devised to accomplish this task automatically. The principles for the ADL to OWL translation were explained in Chapter 4. They were initially designed considering the expressiveness of OWL 1.0.

However, the OWL 2 new features like Qualified Cardinality Restrictions allowed improving the translation method which is now easier to implement as roundabout workarounds are reduced. It should be noted that, apart from being the first step towards clinical rules support and the rest of explained applications, the translation of the ADL semantic and syntactic constraints to OWL restrictions is itself a source of new opportunities for checking logical coherence and validating mechanisms that arises from the inference capabilities of semantic reasoners.

According to the requirements of the clinical research projects described in Chapter 8, the translations principles were implemented for the first purpose of translating the `OBSERVATION` and `EVALUATION` categories of archetypes (currently accounting for roughly 70% of the openEHR repository). After its integration as a module of the *openEHR Java Implementation Project*[108], the translation implementation is being migrated to the new *OWL API*[109], which is focused towards OWL 2. At the same time, the remaining subtypes of archetypes are being integrated in the implementation in order to cover the full range of possibilities of ADL. Further details about the translator implementation can be found in Chapter 5.

While archetypes foster a seamless exchange of clinical data, section 2.1 explains why it is essential to extend such effect to clinical decision support in order to achieve SIOp. This thesis proves that sharing the knowledge expressed in the form of rules is consistent with the philosophy of open sharing and decentralized development, encouraged by archetypes. The two-level approach, introduced in section 2.2, offers a great flexibility to archetype model instances without compromising the interoperability of a unique reference model. Analogously, inference rules and particularly SWRL clinical rules can be specified by means of previously agreed archetypes expressed in OWL. The feasibility of taking advantages from such analogy is studied in Chapter 6, with emphasis on the integration of clinical guidelines.

When translated to OWL, archetypes' definitions can be enriched with SWRL rules and solve the ADL's lack of support for introducing inference content in the definition of archetypes. Any compatible reasoner can then execute the inference and produce data for different purposes. For many years, hospitals and healthcare

---

[108] http://www.openehr.org/projects/java.html

[109] http://owlapi.sourceforge.net/

providers have given very little or no importance at all to interoperability issues when designing their decision support systems. This thesis can be considered as part of a global scale movement towards convincing the entire health community to modify that behaviour. If healthcare systems remain isolated, then the good practice expressed through clinical guidelines and pathways will continue to require up to twenty years to be adopted by the profession as a whole, even when evidence on its favour is unequivocal. The research described in chapters 6 and 8 shows that having inference mechanisms and descriptive knowledge combined under the same syntactic structure provides means for the interoperability of rule systems.

The archetypes and SWRL integration has been successfully evaluated in the research projects briefed in Chapter 8. The collaboration of teaching hospitals like the *Fuenlabrada Hospital* and the *Henares Hospital* has provided the opportunity to test this approach with real clinical data. Technical details of patient's data instantiation and inference execution are given for the *Transverse Sinus Ligation* case study described in section 6.2, and a graphic workflow description of the general *ArchOnt* framework is provided in Figure 38. Although the rest of rules were tested in slightly different environments, it should be noted that they are defined in terms of archetype elements, so they are totally independent of the underlying architecture.

In fact, once the SWRL rules are bound to the OWL version of the archetype, their proper execution only depends on the support provided by the selected reasoner (e.g. Pellet or Jess). Rules can be executed individually, but substantial results come from forward chaining reasoning and rules concatenation as the *Prescription Validation and Allergy Detection* case study illustrated in section 8.1.1. With regard to the limitations of the inference process, there is no specific restriction for this OWL and SWRL integration. Thus, inference boundaries are imposed by the language itself and the reasoner (e.g. the creation of new named individuals and the use of complex mathematical functions, as fractal functions or exponential sums, which are not currently supported by SWRL).

In addition to the value added by the inference capabilities, this research explored a way to have archetypes and SNOMED-CT working together through the OWL interface. The SemanticHEALTH report (V. N. Stroetmann et al., 2009) claims that in order to achieve the overall objective of SIOp, it is imperative that both methodologies and tools are developed with the aim of binding terminologies, EHRs and decision support. As a result, Chapter 7 illustrates a series of benefits from the mappings between archetype concepts and SNOMED-CT terms when establishing semantic OWL connections. This includes improving navigability between clinical ontologies, checking representation correctness and supporting clinical knowledge integration. Nevertheless, offering an OWL environment that provides resources to consistently bind archetypes and clinical terminologies is not enough to guarantee the efficiency of this mapping approach. The success also depends on the quality and accuracy of data mapping to terminology codes, which is still controversial, as stated by Qamar, Kola, and Rector (2008).

As a final summary, the following revision of the thesis's objectives includes references to sections and chapters where the achievement of each objective and their contribution to level 3 of SIOp is described.

1. The openEHR clinical archetypes have been provided with the capability to be expressed through an ontology language, specifically OWL. Chapter 4 studies the different kinds of constraints that can be used within archetypes and provides a methodology to translate them to OWL while preserving the semantics of the original definition.
2. Then, a mechanism has been developed to integrate the OWL version of archetypes with clinical rules. Chapter 6 explains how to combine archetype instances with SWRL rules in order to execute the inference in an interoperable context. Examples of the conclusions that can be reached by such inferential mechanism and the associated benefits are provided in Chapter 8 and section 6.2. In addition to the results gathered from the execution of individual rules, the present thesis has shown the importance of rules concatenation to support clinical guidelines integration in healthcare systems.
3. Full SIOp also demands the integration of clinical terminologies and EHRs. A logical foundation to bind clinical archetypes and terminologies, through an ontology context, has been defined in Chapter 7. An application of such bindings is illustrated in section 8.1 in the *SNOMED-OWL-SWRL* case study that combines the information from three sources

(i.e. SNOMED, EHR and clinical rules) to prevent allergies when prescribing antibiotics.

10 Conclusiones

En la actualidad, numerosos grupos de investigación y empresas del sector médico están considerando varios enfoques para permitir la interoperabilidad entre sistemas heterogéneos de EHR. Como parte de este esfuerzo vemos el desarrollo de grandes terminologías clínicas como SNOMED-CT y modelos de información como el propuesto por la fundación openEHR. Según se explicó en el capítulo introductorio, el objetivo principal de esta tesis es avanzar el alcance de dichos esfuerzos previos y mejorar el enfoque de los arquetipos en la representación del conocimiento clínico por medio de ontologías y lenguajes de la Web Semántica. Por esta vía se busca ofrecer funcionalidades que se corresponden con el nivel 3 de SIOp entre sistemas de EHR. El objetivo principal fue satisfactoriamente alcanzado desde varios ángulos y perspectivas, definidos en cada uno de los objetivos específicos.

Se ha demostrado que es posible integrar los datos de los pacientes, el conocimiento clínico y las terminologías bajo un mismo contexto donde pueden ser procesados automáticamente. Ello permite la interpretación semántica de la evidencia para llegar a conclusiones significativas que pueden a su vez lanzar alertas, recomendaciones de monitorización y ayudas a la toma de decisiones que aumentan considerablemente la seguridad del paciente.

Concretamente, los arquetipos clínicos han sido traducidos a OWL para incrementar la información contenida en ellos por medio de elementos de la semántica computacional. La tesis presenta un método para llevar a cabo esta tarea de manera automática. Los principios de traducción de ADL a OWL se explican en el Capítulo 4. Originalmente se consideró la expresividad de OWL 1.0, sin embargo, las nuevas características de OWL 2.0, como por ejemplo las restricciones cualificadas de cardinalidad (*Qualified Cardinality Restriccitions*), permitieron sustituir ciertos pasos de traducción, algo rebuscados, por soluciones más directas y eficientes desde el punto de vista de la implementación. Se debe tener en cuenta que además de ser el primer paso para el soporte de reglas clínicas y de las restantes aplicaciones descritas en esta tesis, la traducción de las restricciones semánticas y sintácticas presentes en el código ADL a OWL representa por sí misma una fuente de nuevas oportunidades para comprobar la coherencia y crear mecanismos de validación a partir de las capacidades de inferencia de los razonadores semánticos.

De acuerdo con los requisitos de los proyectos de investigación clínica descritos en el Capítulo 8, los principios de traducción fueron primeramente implementados para traducir las categorías `OBSERVATION` y `EVALUATION` de arquetipos (que actualmente constituyen casi un 70% del repositorio de openEHR). Después de su integración como módulo del *openEHR Java Implementation Project*[110] la implementación de la traducción está siendo migrada a la nueva *OWL API*[111], orientada a OWL 2. Al mismo tiempo, los restantes tipos de arquetipos están siendo integrados en la implementación con el objetivo de cubrir todo el rango de posibilidades de ADL. El Capítulo 5 ofrece más detalles sobre la implementación.

A partir del hecho de que los arquetipos clínicos fomentan un intercambio homogéneo de datos clínicos, la sección 2.1 fundamenta la importancia de llevar tal efecto al contexto de la ayuda a la toma de decisiones, como elemento esencial para alcanzar la SIOp. De esta manera, la tesis demuestra la consistencia que existe entre la compartición del conocimiento clínico expresado en forma de reglas y la filosofía de intercambio abierto y descentralizado fomentada por los arquetipos. El enfoque a dos niveles existente entre los arquetipos y el RM es introducido en la sección 2.2. Dicho enfoque ofrece una gran flexibilidad en la definición de las instancias del modelo de arquetipos, sin comprometer la interoperabilidad que se garantiza con un único modelo de referencia. Análogamente las reglas de inferencia, y particularmente las reglas clínicas expresadas en SWRL, pueden definirse en base a

[110] http://www.openehr.org/projects/java.html
[111] http://owlapi.sourceforge.net/

arquetipos previamente acordados y expresados en OWL. El Capítulo 6 estudia la viabilidad de aprovechar tal analogía para la integración de guías de práctica clínica, entre otros elementos del conocimiento clínico.

La traducción a OWL permite enriquecer las definiciones de arquetipos con reglas SWRL, resolviéndose así las limitaciones de ADL para introducir reglas de inferencia en la definición de arquetipos. Una vez combinados el arquetipo expresado en OWL con las reglas SWRL, cualquier razonador semántico compatible puede ejecutar la inferencia y obtener resultados que satisfacen varios propósitos. Durante muchos años los hospitales y centros de salud han prestado muy poca o ninguna importancia a las cuestiones vinculadas a la interoperabilidad de sus sistemas de soporte a la toma de decisiones. Por lo tanto, la presente tesis se considera parte del movimiento internacional hacia el convencimiento de la comunidad médica de que dicha actitud debe ser cambiada. Es vital que se reconozca el hecho de que si los sistemas de salud permanecen aislados, entonces las buenas prácticas expresadas a través de vías clínicas, continuarán necesitando alrededor de 20 años para ser adoptadas de manera generalizada, incluso cuando la evidencia a su favor sea inequívoca. La investigación descrita en los capítulos 6 y 8 demuestra cómo la combinación de mecanismos de inferencia y conocimiento descriptivo bajo la misma estructura sintáctica ofrece los medios necesarios para la interoperabilidad de sistemas de salud basados en reglas.

En la práctica, la integración de arquetipos clínicos con reglas SWRL ha sido satisfactoriamente utilizada en los proyectos de investigación descritos en el Capítulo 8. Gracias a la colaboración de los hospitales universitarios del *Henares* y *Fuenlabrada*, en la Comunidad de Madrid, dicha evaluación se pudo llevar a cabo con datos clínicos reales. La explicación técnica del proceso de instanciación de datos clínicos y de la posterior ejecución de la inferencia se ofrece en la sección 6.2, como parte del estudio de caso *Transverse Sinus Ligation*. Además, la Figura 38 muestra una representación general del flujo de trabajo que sigue el *ArchOnt* framework. Es importante notar que aunque los proyectos donde se han empleado las reglas SWRL presentaban diferentes arquitecturas software, el hecho de que las reglas se definan en base a los elementos de los arquetipos clínicos les permite ser totalmente independientes de dichas arquitecturas subyacentes.

De hecho, una vez que las reglas SWRL se integran con la versión OWL de los arquetipos, una correcta ejecución dependerá exclusivamente de las capacidades ofrecidas por el razonador semántico utilizado (ej. Pellet o Jess). Así, las reglas pueden ejecutarse de manera individual o concatenando la salida de unas con la entrada de otras, es decir, aplicando *modus ponens* repetitivamente en un encadenamiento hacia delante o *forward chaining*. Los resultados más significativos provienen de esta segunda manera, como se muestra en el estudio de caso *Prescription Validation and Allergy Detection* explicado en la sección 8.1.1. Con respecto a las limitaciones del proceso de inferencia, no hay ninguna restricción impuesta por la integración OWL-SWRL estudiada en esta tesis. Por lo tanto, el alcance de la inferencia viene impuesto por el lenguaje en sí y por el razonador semántico (ej. SWRL no soporta actualmente la creación de nuevas instancias como resultado de una regla, ni la ejecución de funciones matemáticas complejas como fractales o sumas exponenciales).

Además del valor añadido a los arquetipos por la capacidad de asociarse con reglas de inferencia, la tesis explora nuevos mecanismos para conectar los arquetipos con terminologías clínicas como SNOMED-CT a través del lenguaje OWL, al cual ambos pueden ser traducidos. En el informe de SemanticHEALTH (V. N. Stroetmann et al., 2009) se considera que para llegar a los más altos niveles de SIOp es indispensable que se desarrollen herramientas y metodologías con el objetivo de enlazar eficientemente las terminologías clínicas, la historia clínica digital (EHR) y el soporte a la toma de decisiones. En este sentido, el Capítulo 7 ilustra una serie beneficios que se obtienen de los *mappings* entre los conceptos definidos en los arquetipos y los términos de SNOMED-CT, cuando se utilizan las capacidades semánticas de OWL para establecer dichas conexiones. La mejora de la navegabilidad entre ontologías clínicas, la validación de la compatibilidad entre diferentes representaciones del mismo concepto y la integración de distintas fuentes de conocimiento clínico constituyen algunos de tales beneficios. Sin embargo, es importante tener en cuenta que ofrecer un entorno basado en OWL que incluye los medios necesarios para enlazar consistentemente arquetipos y terminologías no garantiza la calidad y la precisión del mapeo. La selección de los conceptos a mapear continua siendo un tema polémico, según fundamentan Qamar et al. (2008).

A continuación se ofrece una revisión de los objetivos específicos marcados en la presente tesis, haciéndose referencia en cada caso a las secciones y capítulos que describen el cumplimiento de los mismos y su contribución con el nivel 3 de interoperabilidad semántica.

1. Se ha proporcionado la capacidad a los arquetipos clínicos de openEHR de expresarse a través de un lenguaje de ontologías, en particular OWL. El Capítulo 4 estudia los diferentes tipos de restricciones que pueden utilizarse dentro de los arquetipos y ofrece una metodología para traducirlas a OWL, conservando la semántica de la definición original.
2. Luego, se ha desarrollado un mecanismo para integrar la versión OWL de los arquetipos con reglas clínicas. El Capítulo 6 explica cómo combinar instancias de arquetipos con reglas SWRL con el fin de ejecutar la inferencia en un contexto interoperable. El Capítulo 8 y la sección 6.2 proporcionan ejemplos de los beneficios asociados a este mecanismo así como de las conclusiones obtenidas con la inferencia. Además de los resultados alcanzados con la ejecución de reglas individuales, se ha demostrado la importancia de la concatenación de reglas a la hora de integrar guías de práctica clínica en los sistemas sanitarios.
3. La SIOp exige también la integración de terminologías clínicas con los EHRs. El Capítulo 7 describe los fundamentos lógicos para vincular los conceptos definidos en arquetipos con los términos clínicos organizados en las terminologías, a través de un contexto OWL. En la sección 8.1 se ilustra una aplicación de dichos vínculos en el caso SNOMED-OWL-SWRL que combina la información de tres fuentes (SNOMED, EHR y reglas clínicas) para prevenir las alergias al prescribir antibióticos.

11 Future Work

A long road is still ahead of us in the way to full semantic interoperability. It involves such a large number of changes at both the technical and use case level that only a gradual emergence can be achieved in the best scenario. This thesis has presented a bottom-up approach that supports further development of concrete applications. They should be first implemented in areas of clinical practice that are known to be of high patient safety risk, as well as in priority areas for which the evidence is strongest for a gap to be bridged between current and good practice. The ADL to OWL translation and the rules integration approach constitute a first step toward several semantic tasks related to clinical archetypes that will be addressed as a continuation of this thesis. In addition to the projects mentioned in section 8.3, the most significant are introduced here below.

11.1 Sharing archetypes as linked data

Linked Data is about using Web technologies to create typed links between data from different sources[112]. The OWL representation of clinical archetypes provides the basic semantic web capabilities which are required to link them with other data sources. For example, every element in an archetype can be expressed as a URI. The benefits of Linked Data for clinical archetypes and SIOp include: reducing redundancy, overlapping detection and enabling network effects to add value to

[112] http://linkeddata.org/

data. Having most important repositories and archetype sources linked by this mechanism provides the means to answer queries like: “Is there any archetype definition in any repository related to *Lung Cancer*?” or “Which overlaps are there between archetype repository A1 and A2?”

The first step in order to meet above mentioned benefits will be the implementation of a mechanism to allow clinical archetypes to fulfil the four rules of Linked Data, defined by Berners-Lee (2006). Then, the overall planned architecture will include a RDF triple store like the one provided by the Jena framework[113] that will be feed by the ADL to OWL translation mechanism presented in this thesis. ADL archetypes can be retrieved by a query client from archetypes stores and repositories. Once in the triple store, the RDF representation of archetypes can be shared by an RDF server like Joseki[114] that provides and HTTP interface that fully support SPARQL querying. Finally, this will allow a linked data frontend like Pubby[115] to provide dereferenceable URIs by rewriting the ones found in the SPARQL exposed dataset into the Pubby server's namespace. Such data can be then accessed by a wide variety of existing RDF browsers like Disco[116] or the OpenLink RDF Browser[117], RDF crawlers (e.g. SWSE[118]) and query agents (e.g. SemWeb Client Library[119], SWIC[120]).

11.2 Implementing clinical guidelines and managing archetypes

Several process models have been elaborated to provide support for the specifics of processes in healthcare (Fox, Alabassi, Patkar, T. Rose, & Black, 2006), and particularly to support clinical guidelines (Sutton & Fox, 2003; Mulyar, van der Aalst, & Peleg, 2007). However, Isern and Moreno (2008) offer an analysis and a comparison of eight systems that allow the enactment of clinical guidelines in a

[113] http://jena.sourceforge.net/
[114] http://www.joseki.org/
[115] http://www4.wiwiss.fu-berlin.de/pubby/
[116] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/
[117] http://linkeddata.uriburner.com/ode/
[118] http://www.swse.org/
[119] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/
[120] http://moustaki.org/swic/

(semi) automatic fashion. That analysis concluded that further work is required to integrate such languages in existing healthcare information systems.

Therefore, the research to combine archetypes, ontologies and rules described chapters 6, 7 and 8 will continue to study how clinical archetypes can be used to bridge existing process models and integrate process languages like GLIF in order to encourage full integration of clinical guidelines with the existing healthcare information systems. In this sense, the Map of Medicine[121] may be a very useful source of clinical knowledge as it provides a homogeneous and precise representation of many clinical guidelines.

On the other hand, there is an increasing archetype specification activity that raises the need for techniques to associate archetypes looking for better management and user navigation in archetype repositories. Different computational techniques will be explored in order to generate associations between the OWL version of archetypes, for example, by mapping them to clinical ontologies. A first approach based on mappings to term clusters obtained from the UMLS Metathesaurus is already published by Lezcano et al. (2010).

11.3 The ArchOnt framework scope

With regard to *ArchOnt* framework improvements, current and future research projects (as the ones introduced in section 8.3) will focus on continuing evaluation and assessment of the translation and inference techniques by gathering more results from real clinical environments. Also, methods for modifying and improving the original ADL definitions after processing their OWL representation will be considered (*round tripping*). That includes, for example, consistency checks to help validating archetypes and detecting inconsistent restrictions according to the RM or the specialised parent archetype. According to the last European Commission report about SIOp (V. N. Stroetmann et al., 2009), archetype authoring and validation tools are among the areas needing investment in a medium term.

[121] The Map of Medicine is a collection of evidence-based, practice-informed care maps which connect all the knowledge and services around a clinical condition. The care maps can be customised to reflect local needs and practices by commissioners looking to devise new care pathways. http://www.mapofmedicine.com/

Further steps in the translation development include, on one hand, the analysis of future OWL releases including new structures and built-ins in order to avoid the disadvantages of some of the workarounds available nowadays. For example, the expressiveness problems when translating the ordered lists found in ADL archetypes are due to the current limitations of OWL to express order in a given list.

There are RDF statements that support ordering but modelling information at a lower level than OWL prevents semantic reasoning over such information. The approach introduced by Drummond et al. (2006) shows how currently available OWL constructs can be used to model many aspects of sequences, albeit imperfectly. This approach will be considered for the translation because it allows lists to be checked with a reasoner while statements can be made about indirectly following elements in the list. Other advantages include the capability of expressing *contains* and *does-not-contains*, lists *length* and regular expressions. However the approach does not provide a perfect solution as it is computationally very expensive, memory intensive and difficult to maintain without specialist tools.

On the other hand, future work regarding the translation mechanism will evaluate the compatibility of the presented approach with the CEN/ISO 13606 standard (recently formed as association), as well as the development of an online interface for the ADL to OWL translator as a complement to the already available java libraries.

Versioning clinical archetypes must be considered when a significant modification is performed within a concept definition. Typical modifications include structure simplification, updates, alignment with an altered parent and re-working, among others. Given that the amount of clinical archetypes is continuously growing, dealing with new versions of already existing archetypes is an important issue that will be studied in future work. In this direction, a mechanism will be designed to support the ontology evolution of the archetypes' OWL representation when new ADL versions are released.

12 References

Argüello, M., & Des, J. J. (2008). Clinical practice guidelines: A case study of combining OWL-S, OWL, and SWRL. *Knowledge-Based Systems*, *21*(3), 247–255. doi:10.1016/j.knosys.2007.11.008

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., et al. (2000). Gene Ontology: tool for the unification of biology. *Nature genetics*, *25*(1), 25–29.

Aspden, P. (2004). *Patient safety: achieving a new standard for care*. Committee on Data Standards for Patient Safety, Institute of Medicine, U.S.: National Academies Press.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (2003). *The description logic handbook: theory, implementation, and applications*. New York, NY: Cambridge University Press.

Bail, S., Parsia, B., & Sattler, U. (2010). JustBench: A Framework for OWL Benchmarking. *The Semantic Web–ISWC 2010*, 32–47.

Bangalore, A., Thorn, K. E., Tilley, C., & Peters, L. (2003). The UMLS Knowledge Source Server: An Object Model For Delivering UMLS Data. *AMIA Annual Symposium Proceedings 2003* (Vol. 2003, pp. 51-55). Presented at the AMIA Annual Symposium.

Beale, T. (1999). The GEHR software architecture for a reliable EHR. *Toward an Electronic Health Record Europe* (Vol. 99, p. 328–339).

Beale, T. (2002). Archetypes, constraint–based domain models for future–proof information systems. *Proceedings of the OOPSLA 2002 Conference* (p. 16–32). Seattle, Washington, USA: Northeastern University, Boston.

Beale, T. (2007). *The openEHR Archetype Object Model*. The openEHR Foundation.

Beale, T., & Heard, S. (2007). *Archetype Definitions and Principles*. The openEHR Foundation. Retrieved from http://www.openehr.org/releases/1.0.2/architecture/am/archetype_principles.pdf

Beale, T., & Heard, S. (2008a). *openEHR Architecture Overview*. The openEHR Foundation. Retrieved from http://www.openehr.org/releases/1.0.2/architecture/overview.pdf

Beale, T., & Heard, S. (2008b). *The openEHR Archetype Definition Language*. The openEHR Foundation. Retrieved from http://www.openehr.org/releases/1.0.2/architecture/am/adl.pdf

Beale, T., Heard, S., Kalra, D., & Lloyd, D. (2008). *The openEHR EHR Information Model*. The openEHR Foundation. Retrieved from http://www.openehr.org/releases/1.0.2/architecture/rm/ehr_im.pdf

Beale, T., Heard, S., Kalra, D., Lloyd, D., & Schloeffel, P. (2006). *Introducing openEHR* (No. Technical Report 1.1). The openEHR Foundation.

Berners-Lee, T. B. (2006, July 27). Linked Data. Retrieved February 15, 2011, from http://www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T. B., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, *284*(5), 34–43.

Bicer, V., Kilic, O., Dogac, A., & Laleci, G. B. (2005). Archetype-based semantic interoperability of web service messages in the health care domain. *International Journal of Semantic Web and Information Systems*, *1*(4), 1–23.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, *32*(90001), 267D-270. doi:10.1093/nar/gkh061

Carroll, J. J., & Pan, J. Z. (2006). *XML Schema Datatypes in RDF and OWL* (No. Note NOTE-swbp-xsch-datatypes-20060314). World Wide Web Consortium.

Chen, C., Chen, K., Hsu, C. Y., & Li, Y. C. (2010). Developing guideline-based decision support systems using protégé and jess. *Computer Methods and Programs in Biomedicine*, *In Press, Corrected Proof*. doi:10.1016/j.cmpb.2010.05.010

Chen, R., & Klein, G. (2007). The openEHR Java reference implementation project. *Studies in health technology and informatics*, *129*(1), 58–62.

Christiaens, S. 2006. Metadata Mechanisms: From Ontology to Folksonomy ... and Back. In *OTM Workshops (1)*, 199-207.

Coiera, E. 1997. *Guide to Medical Informatics, the Internet and Telemedicine*. 1º ed. A Hodder Arnold Publication, January 15.

Collen, M. F., Van Brunt, E. E., & Davis, L. S. (1976). Problems of computerization of large computer medical record systems. *Informatics for Health and Social Care*, *1*(1), 47–53.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (1990). *Introduction to Algorithms*. McGraw-Hill Companies.

Day, J. D., Fukushima, T., & Giannotta, S. L. (1997). Cranial base approaches to posterior circulation aneurysms. *Journal of Neurosurgery*, *87*(4), 544-554. doi:10.3171/jns.1997.87.4.0544

Dogac, A., Laleci, G. B., Kirbas, S., Kabak, Y., Sinir, S. S., Yildiz, A., & Gurcan, Y. (2006). Artemis: Deploying semantically enriched Web services in the healthcare domain. *Information Systems*, *31*(4-5), 321-339. doi:10.1016/j.is.2005.02.006

Drummond, N., Rector, A. L., Stevens, R., Moulton, G., Horridge, M., Wang, H., & Seidenberg, J. (2006). Putting OWL in order: Patterns for sequences in OWL. *Proc. 2nd Workshop on OWL: Experiences and Directions*.

Field, M. J., & Lohr, K. N. (1990). Attributes of good practice guidelines. *Clinical Practice Guidelines: Directions for a New Program* (pp. 53-77). Washington, D.C., USA: The National Academy Press.

Forrey, A. W., McDonald, C. J., DeMoor, G., Huff, S. M., Leavelle, D., Leland, D., Fiers, T., et al. (1996). Logical observation identifier names and codes (LOINC) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clinical Chemistry*, *42*(1), 81-90.

Fox, J., Alabassi, A., Patkar, V., Rose, T., & Black, E. (2006). An ontological approach to modelling tasks and goals. *Computers in biology and Medicine*, *36*(7-8), 837-856.

Garde, S., Knaup, P., Hovenga, E. J. S., & Heard, S. (2007). Towards semantic interoperability for electronic health records: Domain knowledge governance for openEHR archetypes. *Methods of Information in Medicine*, *46*(3), 332–343.

Girosi, F., Meili, R., & Scoville, R. P. (2005). *Extrapolating evidence of health information technology savings and costs*. Santa Monica, CA: Rand Corporation.

Golbreich, C., & Imai, A. (2004). Combining SWRL rules and OWL ontologies with Protégé OWL Plugin, Jess, and Racer. *7th International Protégé Conference*. Bethesda. Retrieved from http://protege.stanford.edu/conference/2004/abstracts/Golbreich.pdf

Grimson, J., Grimson, W., Berry, D., Stephens, G., Felton, E., Kalra, D., Toussaint, P., et al. (1998). A CORBA-based integration of distributed electronic healthcare records using the synapses approach. *IEEE Transactions on Information Technology in Biomedicine: A Publication of the IEEE Engineering in Medicine and Biology Society*, *2*(3), 124-138.

Grimson, W., Berry, D., Grimson, J., Stephens, G., Felton, E., Given, P., & O'Moore, R. (1998). Federated healthcare record server–the Synapses paradigm. *International Journal of Medical Informatics*, *52*(1-3), 3–27.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220.

Gruber, T. R. (2009). Ontology. *Encyclopedia of Database Systems*, L. Liu and T. Özsu. Retrieved from http://tomgruber.org/writing/ontology-definition-2007.htm

Guha, R., McCool, R., & Miller, E. (2003). Semantic Search. *The Twelfth International World Wide Web Conference*. Presented at the The Twelfth International World Wide Web Conference, Budapest, HUNGARY.

Heard, S. (2000). The Good Electronic Health Record: trial of a standard healthcare record in Australian general practice. *BRITISH JOURNAL OF HEALTHCARE COMPUTING AND INFORMATION MANAGEMENT*, *17*(1), 24–26.

Heiler, S. (1995). Semantic interoperability. *ACM Comput. Surv.*, *27*(2), 271–273. doi:http://doi.acm.org/10.1145/210376.210392

Horridge, M. (2009). *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. The University Of Manchester. Retrieved from http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_2.pdf

Horrocks, I., Patel-Schneider, P. F., Bechhofer, S., & Tsarkov, D. (2005). OWL Rules: A Proposal and Prototype Implementation. *Journal of Web Semantics*, *3*, 23–40. doi:10.1.1.65.9856

Hwang, S.-K., Gwak, H.-S., Paek, S.-H., Kim, D.-G., & Jung, H.-W. (2004). Guidelines for the Ligation of the Sigmoid or Transverse Sinus during Large Petroclival Meningioma Surgery. *Skull Base*, *14*(1), 21-28. doi:10.1055/s-2004-821356

Isern, D., & Moreno, A. (2008). Computer-based execution of clinical guidelines: a review. *International journal of medical informatics*, *77*(12), 787–808.

Jiang, G., & Chute, C. G. (2009). Auditing the Semantic Completeness of SNOMED CT Using Formal Concept Analysis. *Journal of the American Medical Informatics Association*, *16*(1), 89–102. doi:10.1197/jamia.M2541

Kalra, D. (2008). Terminologies and EHR systems. *Toward Semantic Interoperability of EHR systems: a joint effort for Europe. Expert Workshop*. Brussels.

Kalra, D., Blobel, B., & Regensburg, G. (2007). Semantic interoperability of EHR systems. *Medical and Care Compunetics 4*, 231.

Kifer, M., Lausen, G., & Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, *42*(4), 741-843. doi:10.1145/210332.210335

Kilic, O., Bicer, V., & Dogac, A. (2005). *Mapping Archetypes to OWL* (Technical Report).

Knublauch, H. (2005). *User-defined datatypes in protégé-owl*. Retrieved from http://protege.stanford.edu/plugins/owl/xsp.html

Kuhn, I. M., Wiederhold, G., Rodnick, J. E., Ramsey-Klee, D. M., Benett, S., & Beck, D. D. (1982). Automated ambulatory medical record systems in the US.

Lezcano, L. (2009). A Model for Integrating Disease and Diagnosis Semantics in Clinical Archetypes. *Signs, Symptoms and Findings: Towards an Ontology for Clinical Phenotypes (SSFW09)*. Milan, Italy. Retrieved from http://bimib.disco.unimib.it/index.php/SSFW09_Presentations

Lezcano, L., Sánchez-Alonso, S., & Sicilia, M. A. (2010). Associating Clinical Archetypes Through UMLS Metathesaurus Term Clusters. *Journal of Medical Systems*, *In press*. doi:10.1007/s10916-010-9586-9

Lezcano, L., Sicilia, M. A., & Rodríguez-Solano, C. (2011). Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules. *Journal of Biomedical Informatics*, *44*(2), 343-353. doi:10.1016/j.jbi.2010.11.005

Lezcano, L., Sicilia, M. A., & Serrano-Balazote, P. (2008). Combining OpenEHR Archetype Definitions with SWRL Rules - A Translation Approach. *Emerging Technologies and Information Systems for the Knowledge Society*, Lecture Notes in Computer Science (Vol. 5288/2008, p. 79–87).

Lindberg, D. A., Humphreys, B. L., & McCray, A. T. (1993). The Unified Medical Language System. *Methods of Information in Medicine*, *32*(4), 281-291.

Lowe, H. J., & Barnett, G. O. (1994). Understanding and Using the Medical Subject Headings (MeSH) Vocabulary to Perform Literature Searches. *Journal of the American Medical Association*, *271*(14), 1103-1108. doi:10.1001/jama.1994.03510380059038

Loy, M., & Eckstein, R. (2002). *Java Swing*. O'Reilly Media, Inc.

Markwell, D., Sato, L., & Cheetham, E. (2007). Representing clinical information using SNOMED Clinical Terms with different structural information models. *KR-MED 2008*, 72.

Martin, J., & McClure, C. L. (1985). *Diagramming techniques for analysts and programmers*. Prentice-Hall.

Martínez-Costa, C., Menárguez-Tortosa, M., Fernández-Breis, J. T., & Maldonado, J. A. (2009). A model-driven approach for representing clinical archetypes for

Semantic Web environments. *Journal of Biomedical Informatics*, *42*(1), 150–164. doi:10.1016/j.jbi.2008.05.005

Mei, J., & Paslaru Bontas, E. (2005). Reasoning paradigms for SWRL-enabled ontologies. *Proceedings of International Workshop on Protege with Rules*. Madrid, Spain.

Mulyar, N., van der Aalst, W. M. P., & Peleg, M. (2007). A pattern-based analysis of clinical computer-interpretable guideline modeling languages. *Journal of the American Medical Informatics Association*, *14*(6), 781-787.

Nardon, F. B., & Moura, L. A. (2004). Knowledge sharing and information integration in healthcare using ontologies and deductive databases. *Medinfo 2004: Proceedings Of THe 11th World Congress On Medical Informatics* (p. 62).

O'Connor, M., Knublauch, H., Tu, S., Grosof, B., Dean, M., Grosso, W., & Musen, M. (2005). Supporting Rule System Interoperability on the Semantic Web with SWRL. *The Semantic Web - ISWC 2005* (p. 974–986). Retrieved from http://dx.doi.org/10.1007/11574620_69

Pan, J. Z. (2004). *Description Logics: Reasoning Support for the Semantic Web* (PhD thesis). School of Computer Science, The University of Manchester.

Pan, J. Z., & Horrocks, I. (2005). OWL-Eu: Adding Customised Datatypes into OWL. *The Semantic Web: Research and Applications* (p. 153–166). Retrieved from http://dx.doi.org/10.1007/11431053_11

Pedersen, T. B., & Jensen, C. S. (1998). Research issues in clinical data warehousing. *Proceedings of the 10th International Conference on Scientific and Statistical Database Management* (p. 43–52).

Preston, H. (2000). Information management and technology strategy in healthcare: local timescales and national requirements. *Information Research*, *5*(3).

Pulido, J. R. G., Ruiz, M. A. G., Herrera, R., Cabello, E., Legrand, S., & Elliman, D. (2006). Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, *19*(7), 489-497. doi:10.1016/j.knosys.2006.04.013

Qamar, R. (2008). *Semantic Mapping Of Clinical Model Data To Biomedical Terminologies To Facilitate Interoperability* (PhD thesis). University of Manchester, Faculty of Engineering and Physical Sciences.

Qamar, R., Kola, J., & Rector, A. L. (2007). Unambiguous data modeling to ensure higher accuracy term binding to clinical terminologies. *AMIA Annual Symposium Proceedings*, *2007*, 608–613.

Qamar, R., Kola, J., & Rector, A. L. (2008). Interoperability of Data Models and Terminology Models: Issues with using the SNOMED CT terminology. *KR-MED*. Retrieved from http://ceur-ws.org/Vol-410/Paper09.pdf

Qamar, R., & Rector, A. L. (2007). Semantic mapping of clinical model data to biomedical terminologies to facilitate data interoperability. *HealthCare Computing Conference*.

Rector, A. L. (1989). Helping with a humanly impossible task: integrating knowledge based systems into clinical care. *Hanmu J., Seppo L. (eds). Proceedings of the Second Scandinavian Conference on Artificial Intelligence SCAI-89*, *89*, 560–572.

Rector, A. L. (1999). Clinical Terminology: Why is it so hard? *Methods of information in medicine*, *38*(4/5), 239–252.

Rector, A. L. (2007). *Barriers, approaches and research priorities for integrating biomedical ontologies* (No. D6.1). EC-supported FP6 SA. Retrieved from http://www.semantichealth.org/DELIVERABLES/SemanticHEALTH_D6_1.pdf

Rector, A. L., Johnson, P., Tu, S., Wroe, C., & Rogers, J. (2001). Interface of Inference Models with Concept and Medical Record Models. *Artificial Intelligence in Medicine* (p. 314–323). Retrieved from http://dx.doi.org/10.1007/3-540-48229-6_43

Rector, A. L., Nowlan, W. A., Kay, S., Goble, C. A., & Howkins, T. J. (1993). A framework for modelling the electronic medical record. *Methods of Information in Medicine*, *32*(2), 109-119.

Rector, A. L., Qamar, R., & Marley, T. (2009). Binding ontologies and coding systems to electronic health records and messages. *Applied Ontology*, *4*(1), 51–69.

Rector, A. L., Rogers, J. E., Zanstra, P. E., & Van Der Haring, E. (2003). OpenGALEN: open source medical terminology and tools. *AMIA Annual Symposium Proceedings 2003* (Vol. 2003). Presented at the AMIA Annual Symposium.

Rector, A. L., & Schreiber, G. (2005). *Qualified cardinality restrictions: Constraining the number of values of a particular type for a property*. Retrieved from http://www.w3.org/2001/sw/BestPractices/OEP/QCR/

Rogne, P. (1984). Use of electronic data processing in primary health care. Experiences and thoughts after 2 years' use of a record system for physicians. *Tidsskrift for Den Norske Lægeforening: Tidsskrift for Praktisk Medicin, Ny Række*, *104*(14), 993-995.

Román, I., Roa, L. M., Reina-Tosina, J., & Madinabeitia, G. (2006). Demographic management in a federated healthcare environment. *International Journal of Medical Informatics*, *75*(9), 671–682. doi:10.1016/j.ijmedinf.2006.04.006

Rose, J. S., Fisch, B. J., Hogan, W. R., Levy, B., Marshall, P., Thomas, D. R., & Kirkley, D. (2001). Common medical terminology comes of age, Part One: Standard language improves healthcare quality. *Journal of Healthcare Information Management*, *15*(3), 307-318.

Rossille, D., Laurent, J., & Burgun, A. (2005). Modelling a decision-support system for oncology using rule-based and case-based reasoning methodologies. *International Journal of Medical Informatics*, *74*(2-4), 299-306. doi:10.1016/j.ijmedinf.2004.06.005

Sánchez-Macián, A., Pastor, E., Vergara, J. de L., & López, D. (2007). Extending SWRL to Enhance Mathematical Support. *Web Reasoning and Rule Systems* (p. 358–360). Retrieved from http://dx.doi.org/10.1007/978-3-540-72982-2_30

Schloeffel, P., Beale, T., Hayworth, G., Heard, S., & Leslie, H. (2006). The relationship between CEN 13606, HL7, and openEHR. *Proceedings of HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer*. Seattle, Washington, USA.

Schulz, S., & Hahn, U. (2005). Part-whole representation and reasoning in formal biomedical ontologies. *Artificial Intelligence in Medicine*, *34*(3), 179-200. doi:10.1016/j.artmed.2004.11.005

Schulz, S., Suntisrivaraporn, B., Baader, F., & Boeker, M. (2009). SNOMED reaching its adolescence: Ontologists and logicians health check. *International Journal of Medical Informatics*, *78*, S86–S94. doi:10.1016/j.ijmedinf.2008.06.004

Sheth, A. P. (1999). 0 Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. *Interoperating geographic information systems* (p. 5–29). Norwell, MA: Kluwer.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, *5*(2), 51–53.

Smith, B. (2006). From concepts to clinical reality: An essay on the benchmarking of biomedical terminologies. *Journal of Biomedical Informatics*, *39*(3), 288 - 298. doi:DOI: 10.1016/j.jbi.2005.09.005

Smith, B., & Ceusters, W. (2005). An Ontology-Based Methodology for the Migration of Biomedical Terminologies to Electronic Health Records, *2005*, 704-708.

Spetzler, R. F., Daspit, C. P., & Pappas, C. T. (1992). The combined supra- and infratentorial approach for lesions of the petrous and clival regions: experience with 46 cases. *Journal of Neurosurgery*, *76*(4), 588-599. doi:10.3171/jns.1992.76.4.0588

Stroetmann, V. N., Kalra, D., Lewalle, P., Rodrigues, J. M., Stroetmann, K. A., Surjan, G., Ustun, B., et al. (2009). *Semantic Interoperability for Better Health and Safer Healthcare* (Project report). Luxembourg: Office for Official Publications of the European Communities: European Commission.

Sutton, D. R., & Fox, J. (2003). The syntax and semantics of the PROforma guideline modeling language. *Journal of the American Medical Informatics Association*, *10*(5), 433.

Tu, S. W., Bodenreider, O., Çelik, C., Chute, C. G., Heard, S., Jakob, R., Jiang, G., et al. (2010). *A content model for the ICD-11 revision*. Technical Report BMIR-2010-1405, Stanford Center for Biomedical Informatics Research.

Tzitzikas, Y. T. (2002, September). *Collaborative Ontology-based Information Indexing and Retrieval* (PhD thesis). Department of Computer Science, University of Crete.

Walker, J., Pan, E., Johnston, D., Adler-Milstein, J., Bates, D. W., & Middleton, B. (2005). The value of health care information exchange and interoperability. *Health Affairs*.

Wroe, C. (2006). Is semantic web technology ready for healthcare? *3rd European Semantic Web Conference (ESWC'06)*. Budva, Montenegro. Retrieved from http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-194/paper2.pdf