CROSS-DOMAIN MESSAGE ORIENTED INTEROPERABILITY FRAMEWORK

Cross-Domain Message Oriented Interoperability Framework

By
AZIN DEHMOOBAD SHARIFABADI, B.Sc.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
M.A.Sc.

McMaster University

M.A.Sc. McMaster University, Hamilton, Ontario

TITLE: Cross-Domain Message Oriented Interoperability Framework

AUTHOR: AZIN DEHMOOBAD SHARIFABADI, B.Sc.

SUPERVISOR: Dr. Kamran Sartipi

NUMBER OF PAGES: xi, 79

Abstract

The variety and heterogeneity of information communication standards in different application domains are the main sources of complexity in interoperability provision among those application domains. The maturity of application domains can be assessed by the ease of communication of terms between different stakeholders in the same domain, which is central in defining standards for communication of information among organizations. Currently, most research activities are focused towards standardization and interoperability among information systems within the same domain.

However, an emerging challenge is to address the exchange of information among heterogeneous applications in different domains, such as healthcare and insurance. This requires data extraction to obtain common subsets of information in the collaborating domains. The second step would be to provide intra-domain and inter-domain semantic interoperability through proprietary and shared ontology systems.

In this context, we address the above challenges through description of a framework that employs healthcare standard development frameworks and clinical terminology systems to achieve semantic interoperability between distributed systems among different application domains. A real world case study, which addresses message-oriented integration of business processes between healthcare and insurance is demonstrated.

Dedication

To my beloved family, Shirin, Sohrab, Armita, and Anahita.

To my best friend, Reza, who has been a great source of motivation and inspiration during the very last moments.

To all my gorgeous friends who offered me unconditional love and support throughout the course of this thesis.

Finally, this thesis is dedicated to all those who believe in the richness of innovation.

Acknowledgements

I want to take this opportunity to thank those who helped me to complete this research. I would like to express my thanks to Dr. Kamran Sartipi for his guidance and continuous support. We browsed a wide range of topics and had several discussions to narrow down my research to a concrete comprehensive work. During this research he taught me valuable academic skills that will certainly be beneficial to my future performance.

I would also like to thank the examiner committee members, Dr. Norm Archer and Dr. Douglas Down for their help and time that they put on reviewing this thesis. I also wish to thank many friends and relatives who were always there for me during the hard moments I had.

Contents

1 Introduction 1

1.1 Motivation and problem statement 2

1.2 Proposed solution 3

1.3 Contributions 3

1.4 Thesis overview 4

2 Standards 5

2.1 HL7 5

2.1.1 Refinement process 7

2.1.2 Message structure 7

2.2 Clinical terminologies 8

2.3 ACORD 9

2.3.1 ACORD Life, Annuity and Health Standards 10

2.4 Common Information Model(CIM) 11

2.5 Definition of terms 12

3 Related work 17

3.1 Interoperability in different application domains 17

3.2 Interoperability models 18

3.3 Interoperability infrastructures 20

3.4 Resource mapping 21

3.5 Motivating new approach 22

4 Interoperability Framework 23

4.1 Technical level 26

4.1.1 Over specifying in HL7 v3 27

4.2 Syntactic level 27

4.3 Semantic Level 28

4.3.1 Shared information model 29

4.3.2 Shared terminology system 31

4.3.3 Shared Data Types 31

4.4 Pragmatic Level 31

5 Semantic Interoperability Framework 34

5.1 Collaboration Point Analysis Process 37

5.2 Design Harmonization Process 40

5.3 Dynamic Design Process(HDF): 42

5.4 XML-schema Mapping Algorithm 44

5.5 Harmonize a class with CIM 48

6 Simulated environment 50

6.1 Artifact repository architecture 51

6.1.1 Design Harmonization process: 54

6.2 Message exchange architecture 56

6.2.1 XML family of technologies overview [10] 56

6.2.2 XML Schema [17] 56

6.2.3 XSL/XSLT [13] 57

6.2.4 XPath [14] 58

6.2.5 XSLT processors [41] 60

6.3 Web services 61

6.3.1 Web services [8] 61

6.3.2 WSDL [8] 61

6.3.3 SOAP [9] 62

6.3.4 Document-based web services [43] 62

6.4 Message-oriented interoperability architecture 68

7 Conclusion and future work **72**

Bibliography **74**

List of Tables

4.1 Covering HL7 v3 Transmission Wrapper attributes with Web Service Facilities. 33

6.1 Comparing common message elements to ACORD message elements-scattering measurement 69

List of Figures

2.1 HL7 v3 information refinement process [3]. 6

2.2 HL7 v3 message structure [3] 6

2.3 Acord life party reference model [42] 15

2.4 Common information model (CIM) class diagram [2] 16

4.1 Interoperability levels. 24

4.2 Cross-domain interoperability technical framework based on WS-* family of technologies with minimal use of domain specific knowledge. 26

5.1 Overview of proposed semantic interoperability framework 35

5.2 Collaboration point analysis process detailed steps. Artifacts composing collaboration point analysis model are represented with curly boxes. The steps of collaboration point analysis process are illustrated in regular boxes. 36

5.3 Design harmonization process detailed steps. Artifacts composing design harmonization model are represented with curly boxes. The steps of design harmonization process are illustrated in regular boxes. 38

5.4 Dynamic design process detailed steps. Artifacts composing dynamic design model are represented with curly boxes. The steps of dynamic design analysis process are illustrated in regular boxes. 40

5.5 Application of proposed mapping algorithm to map two schemas. Following is the list of all pairs in candidates HashTable: {Person2, {Person1}}, {isStudent, {person1.EmploymentHistory.Student.EndDate}, ... 44

6.1 Artifact repository architecture. All the artifacts generated in framework processes are stored in a structured repository. 52

6.2 Message exchange architecture. Interaction are implemented as web services. All the messages exchanged among different application domain web services are common message elements. These common message elements will be transformed to each local standard message format by invoking an XSLT processor which is embedded in each web service. 55

6.3 Case Study for the proposed cross domain interoperability among two domains healthcare and insurance. Left: storyboard, harmonization process, concept mapping. Right: details of mapping between two XML schema. . 67

Chapter 1

Introduction

Most IT enabled domains such as banking, government, reservation systems and telecommunication possess standard information models, however, they suffer from vendor's proprietary infrastructure to communicate with other systems. Healthcare domain has already experienced much difficulties in communication among information systems; hence, they have developed a standard way of interoperability through defining comprehensive information and concept representation that allow them to convey a consistent interpretation of semantic concepts.

In this context, we discuss interoperability provision among pre-existing interoperability standards across application domains, according to standard information, knowledge, and services. We propose a framework and corresponding guidelines, that allows us to extract common data and services from participating application domains to perform mutual business processes. The next task is to provide the means for communication of information (syntactic interoperability) and communication of meaning (semantic interoperability). This is achieved through comprehensive and standard common information model (CIM) and concept representations and communication through standard messages.

As healthcare domain is a major domain which invested a huge amount of effort and expert knowledge to build a standard framework to achieve standard way of communication among healthcare systems. This framework is conveyed to obtain a common set of message elements for interoperability; therefore, it is message-oriented. The framework which is followed in healthcare to develop standard documents is named Health Development Framework (HDF). We adopt and modify HDF to suite our problem scope.

The proposed framework consists of three major phases: collaboration point analysis, design harmonization process, and dynamic model design. The collaboration point analysis elicits the requirements for semantic interoperability among the participating domains. The design harmonization process generates common message elements that allow inter-domain communication. Finally, the dynamic model design will elaborate on the interactions among the involving applications.

1.1 Motivation and problem statement

Semantic interoperability within the same domain is a non-trivial research problem, and is a more challenging problem when interoperability is sought among systems in two or more domains.

Some major challenges are as follows: (a) *Collaboration points* between different domains are not clearly and consistently specified by the involving domains. (b) Organizations choose a short term solution for interoperability which is not the best choice. For example, point-to-point pattern proposed by IBM [38] for semantic interoperability is not scalable. (c) Each mature domain has already its proprietary standards which must be maintained after cross-domain interoperability is achieved.

Therefore, we define our research problem as following:

> *Propose a framework to achieve message-oriented semantic interoperability across global application domains such as healthcare and insurance to ensure that design elements in the involving local domains standards are reused and*

the resulting common standard does not conflict or interfere within local information systems.

1.2 Proposed solution

We will start by exploring different levels of interoperability. In this study, we investigate the extent of domain standard usage required to make systems in different domains communicate in the specified level of interoperability. Taking a closer look at healthcare domain, we propose a domain-neutral standard communication in technical and syntactic level of interoperability. The essential common components to achieve semantic interoperability are identified, as well.

Furthermore, a framework is proposed to produce the common design elements and messages among each domain interoperability standard which will reuse local elements from local domain standards. This framework is composed of three phases, namely: collaboration point analysis process, harmonization process, and dynamic design process. An artifact repository structure is proposed to maintain artifacts generated in the designated processes. Also an infrastructure is implemented to obtain transparency among local domain standards.

1.3 Contributions

The contributions of this thesis are as follows:

1. propose an alternative W3C-based standard for transmission wrappers in HL7 v3 messages;
2. augmenting the HL7 development framework;
3. implementing a message exchange infrastructure to exchange common standard messages among application domains;

4. provide transparency among local message standards using the message mapping templates; and

5. Develop a precise XML schema mapping algorithm.

1.4 Thesis overview

The remaining chapters of this thesis are organized as follows:

Chapter 2: briefly introduces message-oriented interoperability standards in different application domains specially Health Level 7 version 3 (HL7 v3) and ACORD.

Chapter 3: addresses the related work and a brief literature review on standard interoperability issued among different application domains.

Chapter 4: represents a general framework for interoperability in different levels and considers the issue of "overspecification in interoperability standards".

Chapter 5: is dedicated to the proposed cross-domain semantic interoperability and information model building framework.

Chapter 6: presents the simulated environment and implementation technologies as well as the real world case study to achieve interoperability between healthcare and insurance.

Chapter 7: wraps up the context and mention the open problems in this context to work on them in the future.

Chapter 2

Standards

Standards are generally required when excessive diversity creates inefficiencies. The healthcare environment has traditionally consisted of a set of loosely connected, organizationally independent units. Patients receive care across primary, secondary, and tertiary care settings, with little communication and coordination among the services. There are many pressures on healthcare information systems to reduce these inefficiencies such that the data collected for a primary purpose can be reused in a multitude of ways.

The healthcare industry has many organizations developing specifications and standards to support information exchange and system integration. These specifications are used to provide interoperability for a wide spectrum of healthcare applications. National and international organizations release standards to effectively integrate healthcare systems. The major standards in healthcare and insurance which are used in this thesis are briefly introduced in Sections 2.1, 2.2, and 2.3. Moreover, several basic concepts for further understanding of the proposed model are identified in Sections 2.4 and 2.5.

2.1 HL7

Health Level Seven (HL7) [3] is an international community of healthcare experts and information scientists collaborating to create standards for the exchange, management

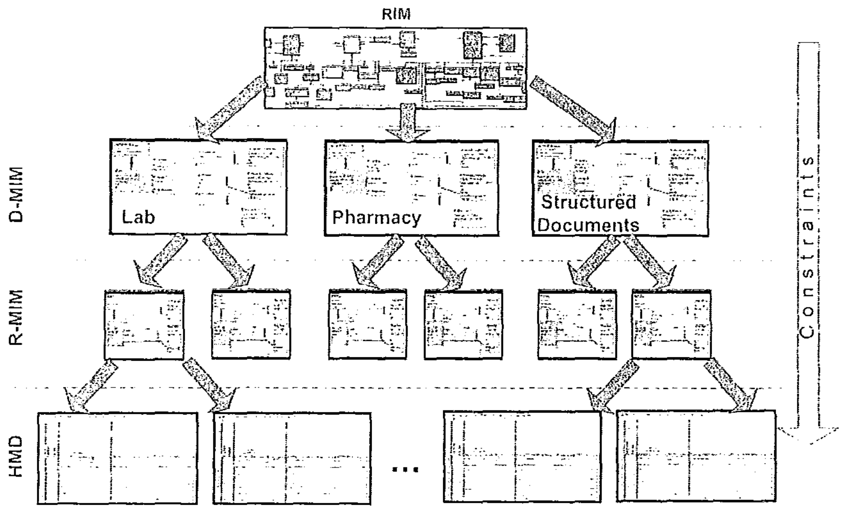

Figure 2.1: HL7 v3 information refinement process [3].

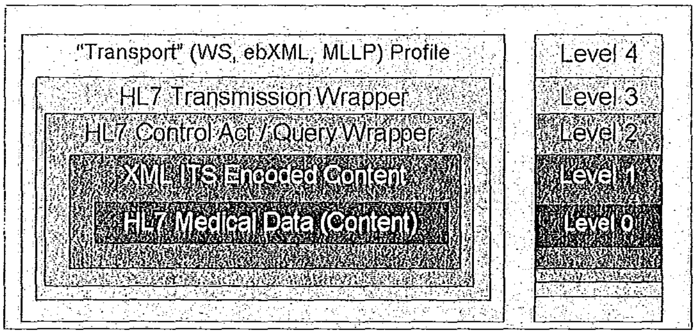

Figure 2.2: HL7 v3 message structure [3]

and integration of electronic healthcare information. HL7 version 3 messaging standard (also called HL7 v3) offers a standard that is testable and provides the ability to certify vendors' conformance. HL7 v3 uses the Reference Information Model (RIM), an object model that is a representation of clinical data and identifies the life cycle of the events that a message will carry. HL7 v3 applies object-oriented development methodology on RIM and its extensions to create messages. A general description of the HL7 standard is beyond the scope of this thesis. However, we do provide here more details on two HL7 concepts: *refinement process* and *message structure* [2, 21].

2.1.1 Refinement process

The strategy for development of HL7 v3 messages and related information structures is based upon the consistent application of constraints to a pair of base specifications, i.e., HL7 RIM and HL7 Vocabulary Domains, and upon the extension of those specifications to create representations constrained to address specific health care requirements. Using the base specifications, the HL7 methodology establishes the rules for refining these base standards to arrive at the information structures that specify a Message Type. Figure 2.1 shows the refinement process specified in HL7 methodology, where the different parts are discussed below.

- *Domain Message Information Model* (D-MIM) is a subset of the RIM that includes a fully expanded set of class clones, attributes and relationships that are used to create messages for any particular domain.
- *Refined Message Information Model* (R-MIM) is used to express the information content for one or more messages within a domain. Each R-MIM is a subset of the D-MIM and contains only those classes, attributes and associations required to compose the set of messages.
- *Hierarchical Message Description* (HMD) is a tabular representation of the sequence of elements (i.e., classes, attributes and associations) represented in an R-MIM. Each HMD produces a single base message template from which the specific message types are drawn.

2.1.2 Message structure

Transactions consist of one or more messages to support both outbound and inbound communications (i.e. send/receive pairs). HL7 has suggested a structure for messages to support transporting interaction information and the actual payload. At the highest level, an HL7 v3 message is composed of two parts (see Figure 2.2):

- *HL7 Transmission Wrapper* includes the information needed by a sending application or message handling service to package and route HL7 v3 messages to the designated receiving applications or message handling services.

- *HL7 Transmission Content* is comprised of two parts:

  - A "Trigger Event Control Act" contains the administrative information about the business event that initiated the sending of this message, who sent it and other associated business information.

  - The "HL7 Domain Content" contains the domain specific content that is specified by the HL7 technical committee to satisfy a use case driven requirement for an HL7 messaging interaction. It includes the core data attributes for the message such as a prescription order or dispense event.

2.2 Clinical terminologies

Clinical terminologies are structured lists of terms which together with their definitions are designed to describe unambiguously the care and treatment of patients. Terms cover diseases, diagnoses, findings, operations, treatments, drugs, administrative items, etc. [4]. A clinical terminology system facilitates identifying and accessing information pertaining to the healthcare process and hence improves the provision of healthcare services by care providers. A clinical terminology system can allow a health care provider to identify patients based on certain coded information in their records, and thereby facilitate follow-up and treatment. Two major clinical terminologies are used in this thesis.

Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) [34] is a comprehensive clinical terminology system that provides clinical content and expressivity for clinical documentation and reporting. SNOMED CT uses healthcare software applications that focus on collection of clinical data, linking to clinical knowledge bases, information retrieval, as well as data aggregation and exchange. The terminology is comprised of concepts, terms and relationships with the objective of precisely representing clinical information across the scope of healthcare. SNOMED covers a semantic network of over 300,000 medical concepts and their relationships. At the top level, there are three main hierarchies (finding, disease, and procedure) and fifteen supporting hierarchies.

2.3 ACORD

ACORD (Association for Cooperative Operations Research and Development) is a global, nonprofit standards development organization serving the insurance industry and related financial services industries. It operates in three areas of the global Industry: personal lines and small commercial; large commercial and reinsurance; and life insurance and reinsurance. ACORD also works with domestic standards bodies where they exist. Standards are a set of processing rules and common information that provide a standard framework for communication with business partners, i.e. a common language spanning the world. ACORD is funded by subscriptions from each of its 500 members.

Standards enable an organisation to develop electronic links with their trading partners to a common method. Without standards each firm would need to develop, build, operate and then maintain different ways of working electronically with each of its trading partners. There is no competitive advantage in having non standard implementations this merely adds to processing complexity. With data, process and communication standards each firm can trade electronically in a standard and therefore more cost effective manner.

ACORD creates standards by working with practitioners in the insurance industry to establish common methods and common information for a particular business function e.g. an accounting transaction. ACORD manages and analyses these activities and creates messages for electronic use by the insurance industry. By moving from paper based systems to electronic the industry can improve speed of service through more efficient processing, achieve better validation of data with improved transparency and reduce barriers to international trade. Basically a more efficient and effective way of working is achieved by individual companies and the Markets within which they operate. The implementation plans for electronic processing of London, the USA and Continental Europe are all based around the use of ACORD messages and standards.

Implementations of ACORD standards: the Market Repositories in London use ACORD standards; the Electronic Placing systems in the USA and the UK also use ACORD standards; and brokers and underwriters alike use the ACORD standards for Accounting, Settlement and Claims processing in the USA, UK and Continental Europe.

The approach ACORD communities are taking is to start small and then to grow. ACORD supports this by defining fundamental core data (skinny messages) as well as the complete data set (fat messages), and by working with each community to communicate roadmaps and schedules for their implementations.

2.3.1 ACORD Life, Annuity and Health Standards

The ACORD Life, Annuity and Health Standards provide the insurance industry with a well-defined vocabulary for expressing insurance concepts in a formally defined specification that enable trading partner to trading partner as well as intra-enterprise sharing of insurance data. The full scope of the Life, Annuity and Health Standards encompasses three primary domains:

Products: All insurance products defined as a financial instrument that have, as a pricing or coverage component, the risk of a person dying (mortality) or becoming disabled (morbidity). This would include Life Insurance, Annuities, Long Term Care, Disability, Health, and other insurance products as well as their supporting investment-related components.

People: All producers or consumers of insurance data including producers/agents, distributors, carriers, reinsurers, regulators, third party service providers, solution providers, and any other users of insurance data. All the business processes where insurance data needs to be exchanged between systems or trading partners, internally or externally, that can be defined and are commonly utilized within the insurance industry.

A public version of the ACORD Life, Annuity and Health Standards are available for use by any interested party and are available on the ACORD web site: http://www.acord.org.

2.4 Common Information Model(CIM)

In this section we provide background knowledge for understanding the CIM which is used in the proposed framework. We have adopted a hub-and-spoke [38] model of semantic interoperability, where a common reference information model is used which consists of the HL7 v3 reference information model (RIM) excluding the healthcare specific classes. This is an evolving reference information model which new classes can be added on demand. HL7 v3 RIM has been chosen for several reasons: (a) Early adopters of the HL7 v3 standards development process have used the RIM to develop HL7-like message specifications in their own environments, therefore it is an accepted information model to be used outside healthcare. (b) Some HL7 member organizations have reported using the RIM as a source of input to their enterprise information architectures or as a starting place for systems analysis and design. (c) The abstract style of the RIM and the ability to extend the RIM through vocabulary specifications make the RIM applicable to any conceivable healthcare system information interchange scenario. In fact, it is conceptually applicable to any information domain involving entities playing roles and participating in acts.

Our proposed CIM is composed of HL7 v3 RIM's three subject areas: "ACTs", "ENTITIes", "ROLEs", and six RIM's backbone classes: "Entity", "Act", "Role", "Participation", "RoleLink" and "ActRelationship" which express the content of business domains. CIM represents business concepts in an abstract way, and the *backbone classes* and their attributes constitute the core part of information model in CIM. These three backbone classes, namely, "Act", "Role" and "Entity", are further refined to represent distinct business concepts, through: i) customizing the backbone classes using inherited sub-classes where additional attributes are used in the sub-classes; and ii) customizing the backbone classes by constraining the attribute values of the backbone class (without adding new attributes) using the notion of *classCode*.

A *classCode* is a distinguished attribute in each of the three backbone classes, whose specific values from a table (namely *controlling vocabulary*) determine different roles for that class (as distinct business concepts) such as: Observation or Examination.

The other three CIM back-bone classes (i.e., Participation, ActRelationship and RoleLink) are customized using a distinguished attribute in each of these classes (namely *typeCode*). These classes represent a variety of concepts, such as different forms of participation or different kinds of relationships between different activities or roles.

The class diagram of CIM is illustrated in Figure 2.4.

2.5 Definition of terms

Below is the description of concepts which are necessary to follow the details in the proposed model. Some of these definitions are directly from HDF 1.2 [39] and some of them are defined independent of any other methodology.

Definitions are as below:

Interaction: any single flow of information between two applications.

Message Element: is any business concept that would be an instance of one of the classes in the three CIM subject areas.

Application Domain: is a group of related business processes which could be performed in a unified organization.

Storyboard: is a narrative description of a series of steps involving some exchange of information between different participants to achieve the objectives of a collaboration business process [39]. An example of a storyboard is shown in Figure 6.3(a).

Technical Expert: is a person with special knowledge and implementation skills in a domain interoperability standard, e.g. in insurance a technical expert is a person who is involved in integration of insurance applications using the ACORD standard.

Domain Expert: is a person with special knowledge or skills in a particular area, e.g. in healthcare physicians are domain experts.

Harmonization: is the process of transforming other glossary of classes to CIM and refinement of messages to acquire common constrained message elements.

Common Message Element (CMET): is a message element generated after applying the proposed framework to interoperate two domains via their standards. The common message elements are used for communication of different domains.

Reusable Element: is an artifact such as an activity diagram for an interaction, or a message scheme which can be used in different business processes properly and eliminates efforts for redesigning and reanalyzing business processes.

Collaboration Point: is a set of related business transactions involving two domain's information systems. (A business process defined between two application domains).

Collaboration Point Analysis Model: is the collection of storyboards, use case models, activity diagrams, message mappings, and business rule descriptions which are generated during collaboration point analysis process.

Domain Standard: is a message-oriented interoperability standard which has the information model of business concept in a specific area such as banking, e.g. HL7 v3. Messages are represented using XML-Schemas[1].

[1]XML schema for message elements is selected since XML is a technology of choice to achieve inter-

Domain Standard Message Element: is a message element from one application domain which is encapsulated in one XML scheme.

Semantic Relationship: is a relationship between two message element attributes or nested element which by making any change in one of them the other one will be changed or at least there is a need to revise it.

Community of Interest (CoI): is the means of categorization of reusable elements in a design process. For each domain there exists one community of interest, e.g. *"diet"* common message element belongs to *"healthcare"* CoI. There is "General CoI" for those message elements which are generic and not domain-specific, e.g. *Person.*

operability among application domains

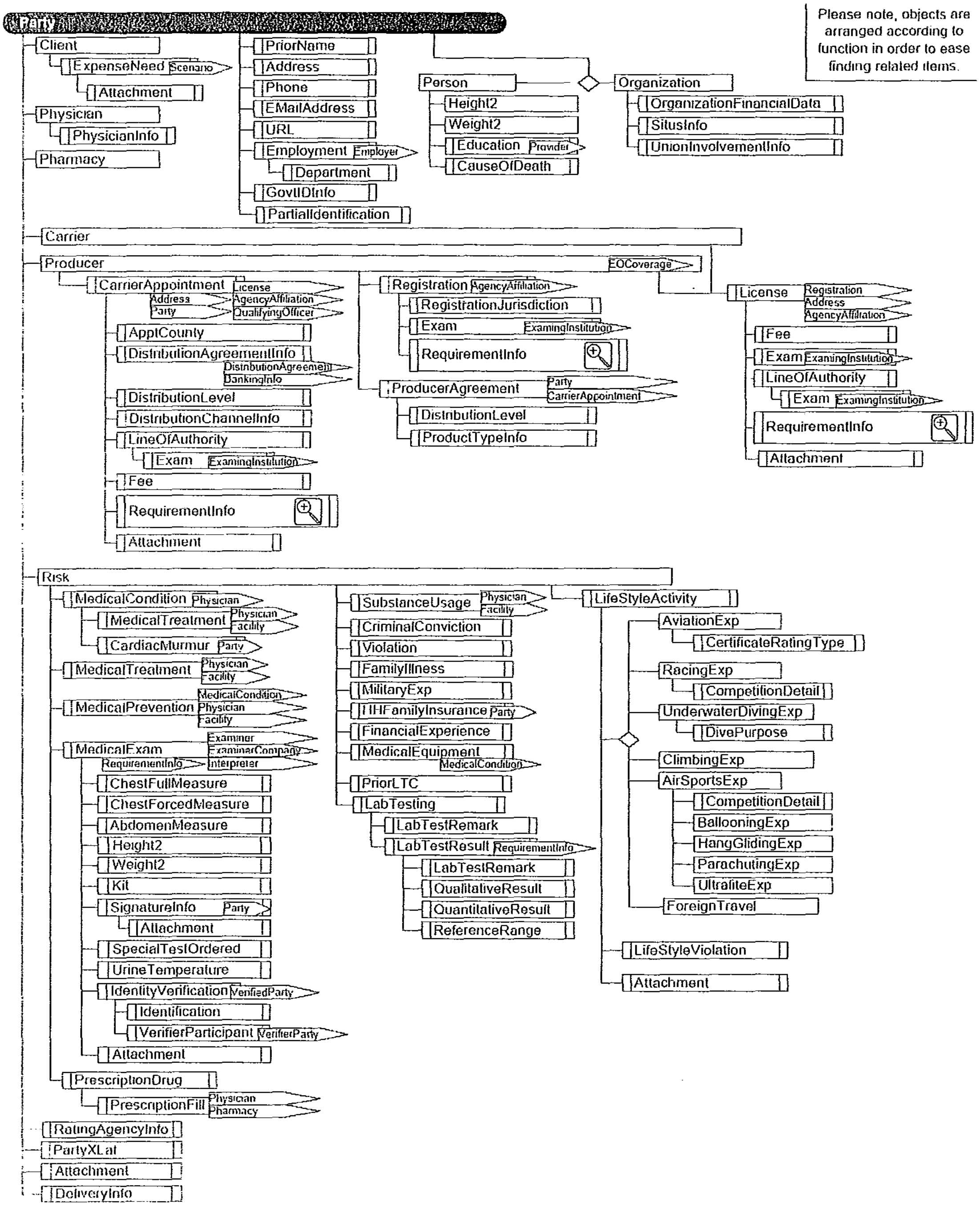

Figure 2.3: Acord life party reference model [42]

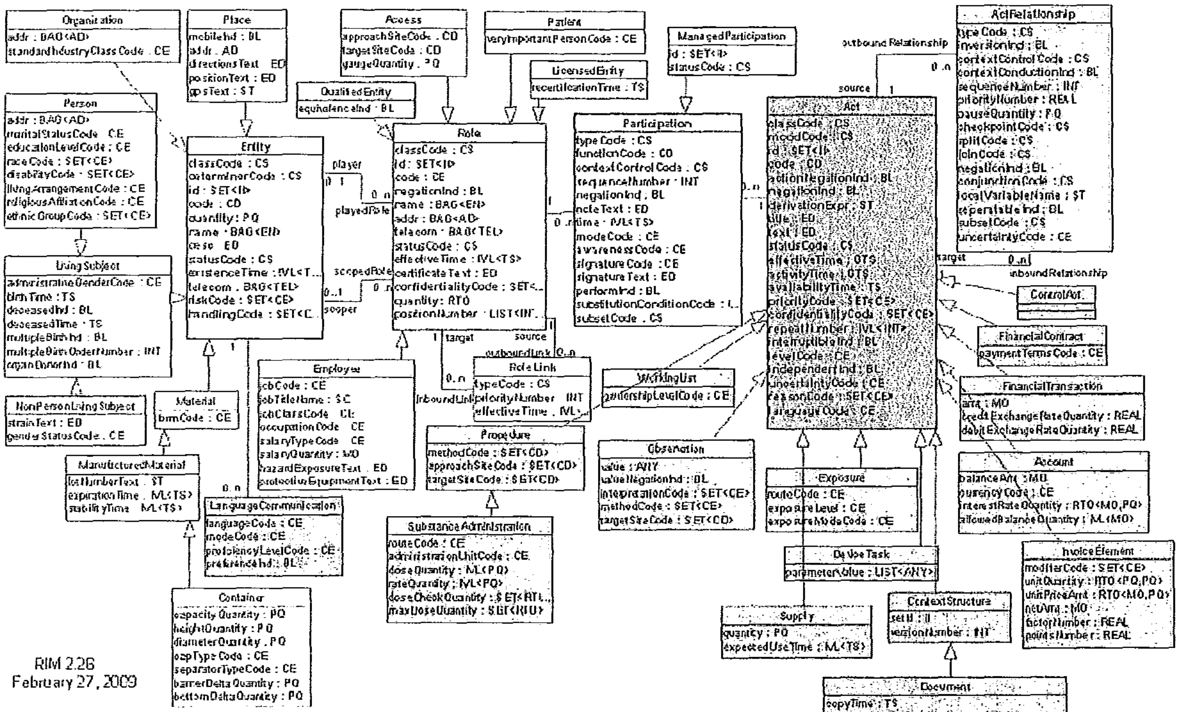

Figure 2.4: Common information model (CIM) class diagram [2]

Chapter 3

Related work

The variety and heterogeneity of information and services in each domain and also the trend towards electronic communication have made interoperability standardization a critical requirement in each domain. From the wide range of interoperability issues, we mostly focus on the semantic interoperability standards and technologies. In this chapter, literature is reviewed for standard models for semantic interoperability, standards for communication of meaning in different application domains and efforts towards eGovernance.

In Section 3.1, we explore several attempts for interoperability standardization in different domains such as airport and healthcare. In Section 3.2, we review existing research initiatives on semantic interoperability models. In Section 3.3, infrastructure analysis for interoperability is addressed. Resource mapping topic is illustrated in Section 3.4 and finally our proposed solution is demonstrated based on the litrature review in Section 3.5 .

3.1 Interoperability in different application domains

Interoperability between heterogeneous systems has been considered in different domains, such as airport, healthcare and military. Airport interoperability standards address the following issues: information exchange model, mapping to database, spatial data standard for facilities, infrastructure and airport environment [6]. Harmonization efforts was undertaken between proposed standards as well. These efforts aim to fill the gaps between these standards and to allow them to work together. Homann et al. [22] propose an interoperability framework for integrating banking systems and present a case study on two European banks using web services. As stated in these articles, all the trends are towards achieving standards in semantic interoperability and different domains are developing their own standards and face the same problems that healthcare has provided partial solutions.

Healthcare already attempted a significant amount of interoperability issues which are faced in other application domains. According to Richesson et al. [37] future directions, we will have lack of purpose for data standards in healthcare domain; a tension is choosing between real solutions to business problems or long-term interoperability. Health development framework [39] propose a methodology to analyze business requirements to generate standard messages for interoperability.

3.2 Interoperability models

A major effort in interoperability research topic is dedicated to analyzing and modeling interoperability requirements. Janssen et al. [31] leverages interoperability and try to address interoperability issues in electronic governance. As stated in this article, interoperability has four levels namely: technical, syntactic, semantic and organizational. They also discuss each levels requirements towards an eGovernment. Guijarro [20] also discusses semantic interoperability for electronic governments. Bourey [35] uses model driven architecture to solve interoperability problem, also De Nicola et al.[33] use a software engineering approach to ontology building which have inspired our work in terms of having iterations of reviews on the message design process. IBM [38] states how to achieve interoperability in a SOA to ensure data consistency and flexibility. They have proposed 4 patterns of semantic interoperability which is point-to-point, hub-and-spoke, Master-data Management and Industry information models semantic interoperability. According to this article, semantic interoperability in a SOA, ensures that service consumers and providers exchange data in a consistent and flexible way. They also introduce patterns and anti-patterns of semantic interoperability. One of the designated anti-patterns are having semantic chaos which everybody define their own schemas and vocabularies and they do not follow any information standards. The other bottle-neck to achieve semantic interoperability is to be overly ambitious about it by planning to face applications and databases to conform to one data model. They mention the major information integration issues to achieve semantic interoperability as data federation, data consolidation and enterprise application integration. Based on the mentioned issues, patterns of semantic interoperability is defined as follows: **Point-to-point semantic integration:** this pattern is recognized as a messy pattern which as data sources grow, its maintenance is not practical.

Hub-and-spoke: a logical data models is defined in this model. This logical model is can be instantiated as a physical federated model.

Master-data-management: this patterns introduces a single version of truth which all the applications and databases should be synchronized with it.

Industry information models: this pattern identifies each domain's interoperability requirement and models their business rules as a focal information model.

All of these models have their own pros and cons which is indicated in the original article for further reference.

3.3 Interoperability infrastructures

There have been vast variety of interoperability infrastructure among information systesm, both vender-dependent and vender-neutral in all levels of abstraction. Motahari et al. [32] propose a conceptual framework for analysing web services interoperability issues. Donachy et al. [16] discuss the requirements for high quality assurance within SOA and grid infrastructures. Hogg et al. [23] proposes and evaluate an architecture for PPS B2B to take advantage of web services technology and states that web services are a proper technology of choice for reuse and minimization of interoperability efforts. There are some limitations that is addressed in these papers and by applying HL7 v3 process to generate standards.

There are efforts to propose architectures and frameworks for interoperability by organizations and software vendors. Oracle's Healthcare Transaction Base (HTB) [5] provides a means to create a comprehensive patient record that can be shared across institutions and geographic regions, so patients can be assured that their medical information are accessible wherever they go. CORBA (Common Object Request Broker Architecture) [1] is OMG's vendor-neutral architecture that computer applications use to collaborate over the networks.

Chen et al. [12] review high level aspects of historical (before 2000) enterprise integration architectures and recent interoperability frameworks and states that there isn't an ideal framework for interoperability yet. This paper addresses SOA, web services and web based technology platforms as outstanding improve in technical interoperability. Shetty et al.[40] address design and development of a large scale autonomic system that uses the concepts of model integrated computing by providing a set of loosely coupled modeling languages that allow the specification of different components of a system. As stated earlier we are going to address interoperability issues on top of an SOA-based infrastructure.

In recent years, the proposed frameworks for interoperability between different systems have been evaluated. Lewis et al. [28] try to identify limitations of interoperability standards. They focus on two areas: semantic and organizational levels of interoperability, and provision of quality of service. Their approach concludes that standards are not enough because of capability of extension and customization and life cycles of standards, and also refer to HL7 v3 as a conflicting standard. In our proposed framework, we adopt a domain-neutral standard that to some extent resolves the above mentioned problems. Mykkanen et al. [29] propose a framework to evaluate interoperability standards; they use a case study of HL7 v3 messaging standard that is defined for scheduling sub-domain. There are some limitations that are addressed in the above two approaches that can be resolved by applying HL7 v3 process to generate standards, other domains can benefit from HL7 v3 development experiences.

3.4 Resource mapping

A critical task in the proposed framework is mapping between different information models derived from each domain standard. There are significant works by Zhao et al.[46] and Kim et al. [26] in entity matching between heterogeneous data sources, XML schema similarity measurement and database scheme matching which are based on classification, text mining, and more specific constrained cascade generalization techniques. Although these approaches are valuable to help information model mapping, precision is the most important issue in our framework and expert domain standard knowledge is required. Therefore, we can not use these techniques to improve the mapping precision.

Recently a proposal by OMG group [19] is provided. They intend to map two messaged from different messaging formats. First, they remove the syntax from the messages and convert it to a set of business elements. To transform the message from one format to another they will do the reverse procedure. Their effort is towards building a hub-and-spoke interoperability model and develop a “standard framework and methodology” to address the mapping problem in finance industry. Many novel concepts in message mapping are introduced in this proposal.

3.5 Motivating new approach

In our approach, we present a framework to solve each domain's standard inconsistency issues by generalizing HL7 v3 development process. Our proposed framework is a simplified and modified version of HDF and suggests a hub-and-spoke pattern of semantic interoperability. This framework aims at long-term solutions besides the fact that we satisfy the short-term objectives as well; therefore it will preserve both point-to-point and hub-and-spoke patterns pros as well as maintaining industry models in each individual domain.

Proposed framework is build upon a service oriented architecture. In contrast to proposing different vendor-based products for interoperability, we recommend to use web services which are globally accepted and allow the users to set up low-cost networks to join. Web services are the best technology of choice for interoperability as stated in the reviewed articles.

Chapter 4

Interoperability Framework

Due to increasing popularity and adoption of distributed systems, heterogeneity has become a major issue in interoperating among existing enterprise applications. Systems may be distributed in terms of data, computing and users, hence there would be several advantages to allow stand-alone systems to interoperate via well-defined services and data. Most of these systems are multilingual and have been developed in different platforms so interoperability between them is a major challenge. However, due to geographical distances, integration of such systems into a monolithic system is not an option anymore. In this context, standardization is a key requirement for providing interoperability at different levels of communication hierarchy which prevents conflicts such as overlap, incompatibility and mismatching. As shown in Figure 4.1, interoperability among information systems is separated in four levels. Standards which facilitates interoperability in these four levels should be independent as much as possible. These standards may be either domain-specific or domain-neutral; in either case, there should be a proper mapping between different standards within different levels of interoperability to ensure effective operations. Developing domain-neutral standards will allow cross-domain interoperability among applications from relevant domains such as banking, insurance and healthcare to collaborate and maintain the quality of services across the domains.

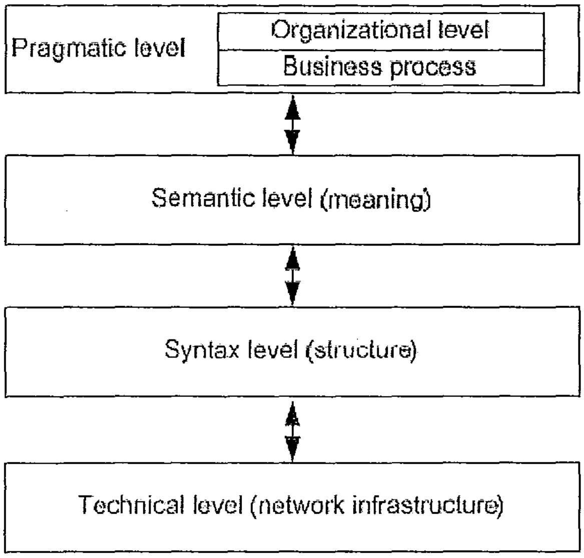

Figure 4.1: Interoperability levels.

Within the application domains that have embraced IT for decades, including: banking, government, reservation systems and tele-communication, most legacy systems communicate through vendor's proprietary process with no standard representation of information and messages. On the other hand, domains such as healthcare that have already experienced much difficulties in communicating medical and clinical terminologies, have developed comprehensive information and concept representations that will allow consistent interpretation of concepts among heterogeneous legacy and new healthcare systems. Within such a development framework, the domain information undergoes a sequence of refinements from a comprehensive body of knowledge representation (as class diagram) down to interoperable concepts and terminology hierarchies that are understandable by all relevant stakeholders within the same domain. Similarly, standard functionality and operations within the domain are incrementally refined from task scenarios down to a collection of standard messages that will be consequently populated by the above standard concepts and terms to interoperate. In such a generic message development framework, service oriented architecture is perfectly applicable to provide the necessary abstraction at the business rule level while maintaining the standard and vendor-independent lower-level technologies such as web services that warrant seamless interoperability at different granularity levels. In this context, task forces in different application domains (such as healthcare) have developed their own set of standards for interoperating at business rules to low-level communication protocols which hinders further interoperability provision across other domains such an insurance and banking.

In this chapter, we address such a problem, namely "*over-specifying domain specific interoperability standards*" and propose a framework to design a cross-domain interoperability standard based on a minimal amount of domain-specific knowledge during communication between applications in two relevant domains. As a comprehensive example, we will consider HL7 v3 messaging standard in healthcare domain and will address different interoperability levels according to our framework by using web services incorporated to HL7 v3. We also present detailed arguments in adopting web services standard transmission infrastructure instead of some specifications in HL7 v3 messaging standard.

This chapter has been organized as follows.

Figure 4.2 illustrates our proposed framework for cross-domain interoperability among closely related application domains based on WS-* family of technologies. In this context, standards are needed at each level of interoperability, namely technical, syntactic, semantic and pragmatic. These standards could be either domain specific or domain neutral. The framework is intended to minimize the use of domain-specific standards that traverse different interoperability levels. This minimization of domain knowledge facilitates interoperation among systems in different domains due to requiring low domain expert knowledge to develop an interoperability middleware. Thus, domain-neutral standards are needed to cover as many levels of interoperability as possible. In the followings, we describe the four levels of interoperability within our framework. We have also provided

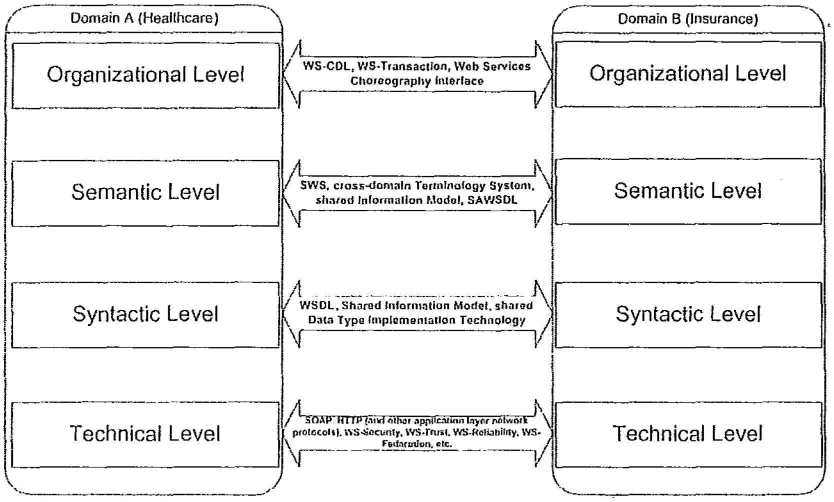

Figure 4.2: Cross-domain interoperability technical framework based on WS-* family of technologies with minimal use of domain specific knowledge.

two case studies, one for technical level and one for semantic level of interoperability, with refer to two standards ACORD and HL7 v3 in insurance and healthcare domains, respectively.

4.1 Technical level

The technical level refers to the data transportation aspects such as: security, reliability, and authentication. There is no need to apply specific domain knowledge at this level. For example, WS-* family have a set of specifications for messaging that are developed and widely used by industry and can be applied to cover all the required message passing requirements. As shown in Figure 4.2, existing protocols such as SOAP and HTTP can handle technical interoperability and WS-* family specifications can be developed on top of SOAP to add more capability to message passing. In the followings, we will discuss to what extent technical issues have been covered by domain specific standards. In this case, HL7 v3 transmission wrappers are studied and we conclude that these wrappers have been completely handled by the WS-* family technology, and hence it is efficient to leave such responsibilities for WS-* family to implement message passing technical infrastructure.

4.1.1 Over specifying in HL7 v3

In our approach, "*over specifying*" refers to a case where a domain-specific standard violates its defined boundaries by specifying standards of another interoperability level. We argue that HL7 v3 (as a domain specific standard) should not specify the requirements for technical level of interoperability infrastructure for message transmission between systems. However, such an "over specifying" makes it difficult to use these message types in service oriented architectures. In HL7 v3 messaging, transmission wrappers are the outermost layer and they have been designed to cover transmission issues such as: acknowledgement messages, packaging and routing messages, identification of sender and receiver, transport specifications, and attributes that address the message handling of the receiver counterpart. These transmission wrappers consist of the attributes shown in Table 4.1. However, all these features (except those related to payload of the transmission) can be handled by the corresponding web service protocols and WS-* family facilities. Table 4.1 represents a goal-based mapping between WS-* family facilities and classes provided by the transmission wrappers.

4.2 Syntactic level

At this level of interoperability, the main concern is the structure and format of the data that are exchanged. The abstract data types specified by HL7 v3 documents [2] are used in our framework, and ISO (International Standard Organization) data types are used as an implementation of these abstract data types. The XML format that is used to generate messages is specific to XML platform and the message contents are derived from serialization of each message content in cross-HMD step of refinement that is defined in the semantic interoperability. The XML schema is generated from the tabular representation of the standard message contents which allows both the sender and receiver to refer to the same schema by the means of the standard message identifier. The detailed implementation specification of data types can be found in [2].

WSDL standard are applied both to expose services and to define signatures that should be used to invoke the services among the SOA components. WSDL is an XML document that describes services in terms of a series of communication endpoints and ports to expose them to service consumers. Abstract definitions of service components are separate from their concrete network protocols and data format binding. This binding is the focus to achieve syntactic interoperability. This feature is illustrated in the syntactic level of Figure 4.2. Therefore, at the syntactic level the required domain-neutral interoperability has been provided by the communication abstraction of web service description languages (WSDL) which provides a common method of accessing the required domain-specific data types and a common grammar to parse the XML messages.

4.3 Semantic Level

At this level of interoperability, domain specific knowledge is widely needed. We propose a framework to ensure semantic interoperability between systems in different application domains with minimal effort to use each domain's specific knowledge or standard. At this level of interoperability, we require an information model, a terminology system, and a shared set of data types. Our framework for interoperability follows the Hub-and-spoke pattern [38] as opposed to existing point-to-point solutions. Point-to-point patterns are complicated and inefficient when data sources grow over the time. In the followings, each component of this framework is discussed in detail:

4.3.1 Shared information model

In order to achieve cross-domain interoperability, we propose to use the same process of refinement as in HL7 v3 information model [11] to build a consistent information model between different domains. We adopt a Core-RIM that represents the common set of classes, attributes and relationships between classes among all the existing domains. The Core-RIM is derived from HL7-RIM and consists of classes that are not specific to healthcare. Examples include: *WorkingList*, *Procedure*, and *Exposure*. For each set of scenarios to perform information exchange between two domains, there exist a cross-DMIM which is a clone of classes of Core-RIM that are constrained to the requirements of that set of scenarios. Further refinement is performed to generate cross-RMIM for each transaction of a scenario and cross-HMDs for their required interactions. This framework has a bottom up approach, where the steps for building the shared information model are discussed below.

i) Scenario definition: first we describe a set of scenarios that require data flow between the domains. For example: "the insurance domain may want to receive pharmaceutical information of a person from a pharmacy". This step should generate a set of use case diagrams.

ii) Transaction extraction: for each scenario we extract its use cases as separate transactions, where each transaction can be represented by an interaction diagram.

iii) Interaction extraction: each transaction is realized by one or more interactions, each of which is a single data flow from one application to another.

iv) Information definition: for each interaction we define a set of information to be exchanged. These information should be restricted only to those needed for that specific data exchange. The output of this step is a set of class diagrams, attributes and their relationships.

v) Mapping to Core-RIM: the information from the previous step will be mapped to the Core-RIM classes, attributes and relationships.

vi) Extension points: after the mapping step, if there is any further information remained (i.e., classes, attributes or associations), we extend the Core-RIM and generate a new class diagram for RIM. For the new generated classes that should be specializations of the foundation classes of RIM (namely, Act, ActRelationship, Role, RoleLink, Participation and Entity or the subclasses) we find the type of the attributes from the shared data types. As an output a new class diagram for Core-RIM is provided.

vii) Cross-DMIM: for each pair of application domains (e.g., healthcare-insurance) we clone all classes that are needed to communicate between the domains and perform a refinement in terms of cardinality, relationship names, etc.

viii) Cross-RMIM: for each transaction extracted in step ii, we develop an R-MIM and finalize the information that we should put in each message for each interaction.

ix) Serializing the information for each message (cross-HMD): in this step we develop a tabular representation of data in each message independent of any implementation technology.

x) Generate message schemas: according to the tables produced in previous step, message schemas for each interaction is generated.

xi) Mapping: a mapping between the cross-domain information model to domain-specific information model is provided to make the system process the received information properly.

In this framework two steps *iv* and *v* are meant to minimize domain-specific knowledge. We have a Core-RIM that guarantees all the produced messages are derived from the same information model. The refinement process for each scenario (also can be considered as information categorization for cross-domain interaction) is used to manage vocabularies, class associations, and mandatory attributes in each interaction. For a detailed description of refinement process refer to HL7 v3 Ballot [11]. Future trends may use this information model to semantically annotate WSDL and expose it to have a complete set of semantic and syntactic interoperability.

4.3.2 Shared terminology system

The shared terminology system possesses the same architectural style as SNOMED CT terminology system. It consists of concepts that are logically defined by relationship to one or more other concepts. Formal rules for *post-coordinated* expressions are used to make this terminology system precise in terms of relationships between concepts. Any concept can be refined using this formal rule. Concepts are represented in a *compositional grammar* [34].

In our case study to achieve exchanging pharmaceutical information across two different systems in different domains, we accepted SNOMED CT vocabulary system architecture and added concepts needed to be exchanged to the whole terminology system. To expand terminology system to include insurance specific concepts, we also added concepts that are used for exchanging pharmaceutical information in ACORD *Life and Annuity Standards Licensing and Appointments Implementation guide V2.1* Lookup section [42].

4.3.3 Shared Data Types

To have a meaningful data exchange, definition of the values that are exchanged is inevitable. Any data element within a data flow between two systems has a data type. HL7 v3 messaging standard uses a complete set of external data type systems and different implementation technologies can be employed as mentioned in HL7 v3 Ballot [2]. For our case study we used HL7 v3 data type system (as shared data types) due to its comprehensive coverage of all data types defined by ACORD.

4.4 Pragmatic Level

At this level, domain specific knowledge is required to provide a set of business processes between two or more domains. Furthermore, these business processes can be specified using WS-* family facilities listed in Figure 4.2 such as BPEL. Anzbck et al. [36] provide a semi-automatic tool to generate these web services and BPEL files. To have pragmatic interoperability at the organizational level, Service Level Agreements (SLAs) are used to define responsibilities and agreements between industries to use their mutual services.

HL7 v3 Transmission Wrapper Attribute	Goal	Web Service Facility
id	Transmission Identification	WS-Addressing
creationTime	The Transmission Creation Time	WS-Security
interactionId	Unique Interaction Identifier	WS-Addressing
responseModeCode	Explanation of Required Response Time	WS-Addressing
securityText	Extra Security Features of a	WS-Addressing and WS-Security
versionCode	Transmission Identification	WS-Addressing
Sender	Identification of the Sender	SOAP
Receiver	Identification of the Receiver	SOAP
respondTo	Identification of the responsible application role	SOAP
AttentionLine	Representation of Technology Specific Data	N/A
ControlActProcess	Body of the HL7 v3 Message	Message Body
AttachmentText	Payload Attachments	SOAP-Attachments
typeCode	Acknowledgement Details	WS-Reliability
code	Acknowledgement Details	WS-Reliability
text	Acknowledgement Details	WS-Reliability
location	Acknowledgement Details	WS-Reliability

Table 4.1: Covering HL7 v3 Transmission Wrapper attributes with Web Service Facilities.

Chapter 5

Semantic Interoperability Framework

In this chapter, a framework to build common components to achieve semantic interoperability is demonstrated. Having different standards in different domains, they require a compatible and comprehensive way of semantic interoperability. Towards this purpose, a common information system, a common terminology system and a common set of datatypes and value sets, adapted among local application domains, are necessary. The output of the framework is a common information model and a code-based post-conditioned terminology system which will cause transparency among local domain standards. This information model will evolve based on future demand for communication of application domains.

Following are two significant features of this framework: i) This framework is reusing message elements modeled in each local domain standard, therefore, remodeling business concepts is avoided. Message elements of the participating domain standards are extracted and harmonized with CIM. Afterwards, they are refined to acquire constrained common message elements. ii) Our approach covers both process-oriented and data-oriented modeling. *Dynamic design models*, which covers processes, address process-

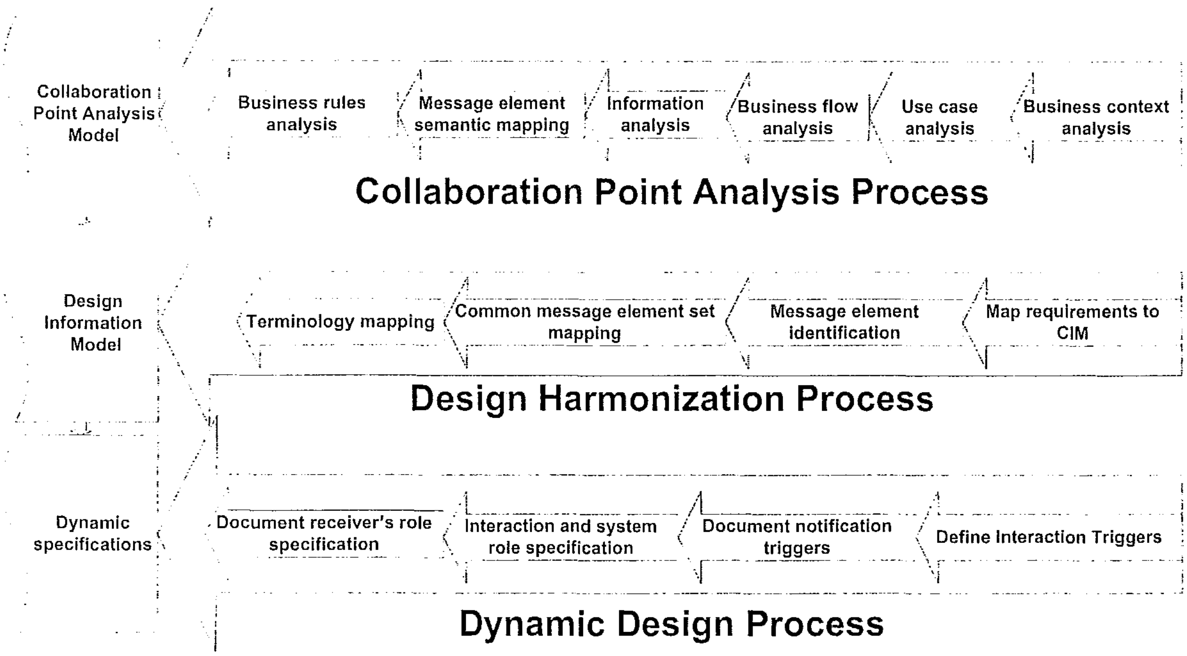

Figure 5.1: Overview of proposed semantic interoperability framework

oriented modeling and *common information model*, which covers data modeling, represents data-orinted modeling.

Cross-domain interoperability business and requirement analysis in the proposed semantic interoperability framework is performed by two groups of technical and domain experts from both involving domains. Health Development Framework (HDF) is adapted as a standard methodology to develop common message elements. Several steps in its general processes are added or modified and the remaining ones are derived from HDF directly. The framework has 3 processes, namely, "Collaboration Point Analysis Process", "Design Harmonization Process", "Dynamic Design Process".

All the process steps are discussed in detail in this chapter. These steps are divided into two groups: not modified HDF steps and embedded steps which are distinguished by "HDF" and "proposed" tags respectively, in the following sections. The "HDF" steps are the same as they are stated in HDF documents, therefore, for detailed instructions

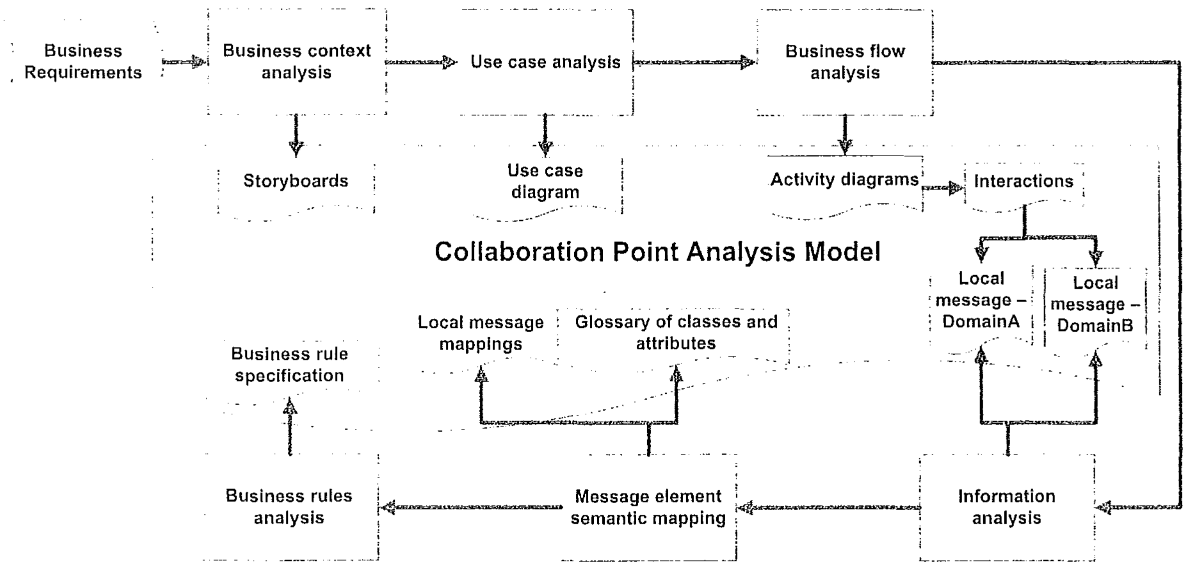

Figure 5.2: Collaboration point analysis process detailed steps. Artifacts composing collaboration point analysis model are represented with curly boxes. The steps of collaboration point analysis process are illustrated in regular boxes.

refer to [39]. The "proposed" tagged steps are a modified or completely new steps added to the HDF domain analysis process and harmonization process.

These processes and their steps are illustrated in Figure 5.1.

In the collaboration point analysis process, the business requirements of interoperability among two domains are analyzed in terms of activities and roles participating in these activities. The outcome of this step which is *collaboration point analysis model* is design harmonization process input. The design harmonization step takes the CIM as another input and harmonizes the requirements with CIM (static information model) and produces a design information model and a set of common message elements for each information exchange requirement. Dynamic design process is performed to add detailed specifications to design information model.

5.1 Collaboration Point Analysis Process

In this section we discuss the collaboration point analysis process which results in collaboration point analysis model and generates artifacts namely storyboards, use case diagrams, activity diagrams, and business rules specifications(optional). The overview of this phase is shown in Figure 5.2.

Business context analysis (HDF): Provision of a proposal for collaboration is the first requirement to initiate a new iteration of this step. In this step, business context, system behavior and information exchanged among systems are extracted and recorded in a storyboard.

The detailed steps to analyze the business context, where interoperability is required, are as follows: a) a proposal to add a standard specification for communication between two application domains is provided. b) the business requirements are derived from the proposal. c) actors in the contained business process are identified. These actors may be human actors, organizations, and systems. d) the information exchanged during the business process and flow of events will be recognized. e) pre-conditions of information exchange is described.

Use case analysis (HDF): the use case analysis describes typical scenarios of end-users interacting with systems for the purposes of sharing or looking up information. The storyboards will be analyzed and use case specifications are developed as outputs. Use case models formally identify actors and use cases illustrated in the previous step and participation of actors in associated use cases in functional areas. Each use case focus on achieving one business goal or task.

Detailed steps are as the following: a) storyboards are analyzed to identify the systems, business actors, functions and actions performed. b) describe the conditions of information exchange and; c) identify the responsibility of the receiving actor.

Business flow analysis (HDF): in this step, each storyboard is refined and represented as an activity diagram. The diagram will illustrate the the activities and flow of

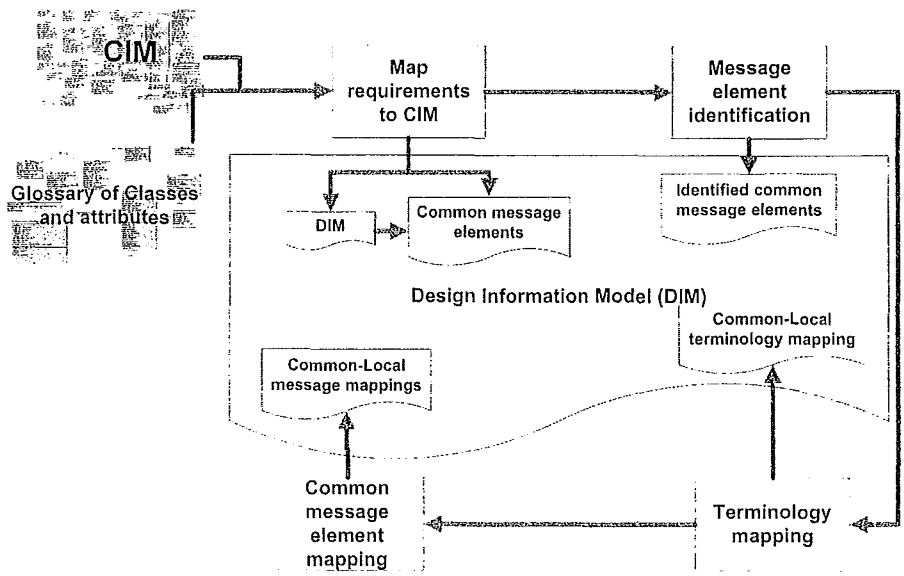

Figure 5.3: Design harmonization process detailed steps. Artifacts composing design harmonization model are represented with curly boxes. The steps of design harmonization process are illustrated in regular boxes.

a business process between two domains. These activities are referred to as transactions. Each transaction is composed of several interactions. Each interaction is a single flow of information between two participants. For each interaction, the information which is required to be exchanged will be identified and written in plain text.

The detailed steps description is as the following: a) all the steps in the related use case model is clarified and expanded. b) the flow of the business process is visualized in an activity diagram. c) the process flow which is illustrated in the activity diagram is divided into steps d) the information exchanged is defined clearly with the related business triggers.

Information analysis (proposed): this step should be performed by each domain's technical expert team individually. The message element set which is required for each interactions is extracted without redundancy and ambiguity. These message elements model the syntactic and semantic relationships among business concepts in a business process and responsible parties and entities. The semantic meaning of each message element and attribute is documented as well.

All of the selected message elements in the message element set is tagged to indicate that these message elements are selected once and mapped to common message elements.

The detailed description of steps are as following: a) specify the message of interest for each interaction. b) value sets are analyzed for each field in the message element set. The domain's local value sets and code systems are identified and analyzed as enumerations in the the message schemas. c) both domain's technical experts review the message requirements to finalize the message element set deliverables to the next step.

Message element semantic mapping (proposed): each group finalize the list of message elements in their local standard for each interaction. This step is the entry point for merging two local standards together and generate common standard messages. The message mappings are performed to gather all the information agreed between two domains consistently and avoid producing redundant design elements during harmonization. One of the message element sets are selected as a source data and the other is the target set. Based on the procedure defined in Section 5.4 the source message element set is mapped to the target set and the target set is ready to be harmonized with CIM. All of the message elements in the target set will be transformed to class diagrams with corresponding attributes and associations.

Business rules analysis(HDF): this step is for further accuracy and precision. To meet semantic interoperability requirements, additional specifications are necessary to add business triggers. In all the cross-domain business processes, there are necessary business rules associated with each message exchange.

Detailed description on the step is as the following: a) all the conditions on data exchange in each activity diagram should be analyzed. b) the result of this analysis is written in a narrative plain text. c) after domain expert reviews, it is formally specified

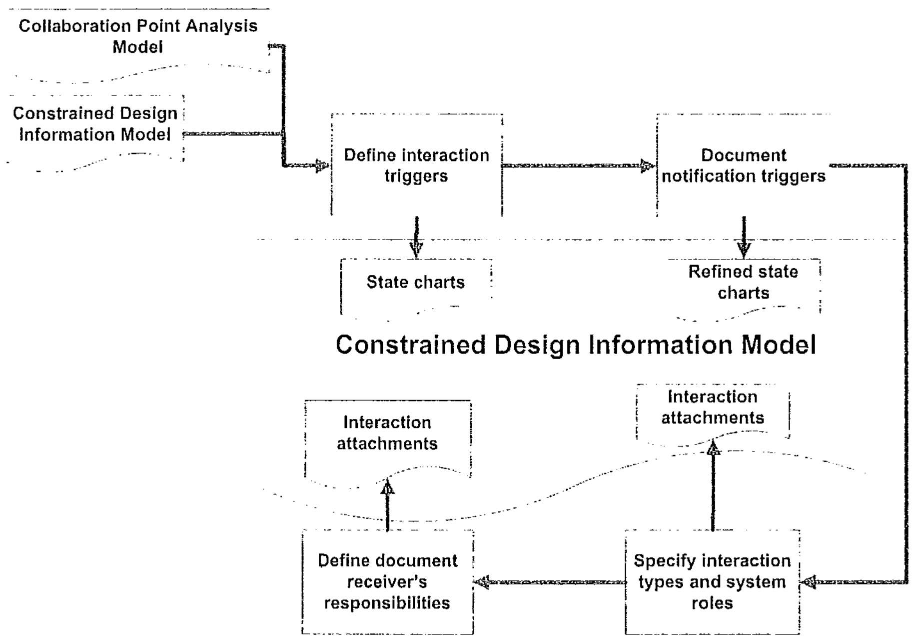

Figure 5.4: Dynamic design process detailed steps. Artifacts composing dynamic design model are represented with curly boxes. The steps of dynamic design analysis process are illustrated in regular boxes.

in BPEL.

All the artifacts from previous steps are shown in Figure 5.2. These artifacts will compose the collaboration point analysis model. The artifacts are stored in a structured document format and will have the attributes of a structured document. These attributes are illustrated in next chapter under formal definitions.

5.2 Design Harmonization Process

In this section we will review the steps to harmonize the collaboration point analysis model with CIM (the static back-bone of our framework)as shown in Figure 5.3. The result of this process is a design information model which is a set of CIM class clones. The collaboration point analysis model is mapped to CIM to identify which classes from

CIM are required for each interaction. In some cases it is necessary to include multiple clones of the same CIM class to model different parts of the analysis model. Each class derived from a CIM class is given a unique name that is representative of its business use.

Map requirements to CIM (proposed): the glossary of the classes and attributes from collaboration point analysis model is the input of this step. The classes, attributes and associations should be mapped to the CIM.

The detailed declarations on this step is as follows: a) the object of interest for this interaction is determined. The object of interest is the root element of the corresponding XML message from local standard and the flow of interaction is based on the status change of this object. b) the object of interest is mapped to its corresponding class in CIM using the procedure defined in 5.5. c) all the classes which are in association with the focal object are mapped to CIM using the same procedure as well. d) after this mapping, all the classes are put together and create a Design Information Model (DIM) for the specified interaction or a set of similar interactions with the same data exchange requirements. e) the design information model is reviewed to apply the constraints to the vocabulary domains and data types. f) the produced DIM is published for a set of interactions which can use it.

Generating common XML message elements from DIM: In this step, all the classes in DIM are serialized as an XML message element.

The detailed description of this step is as follows: a) the object of interest in DIM is selected as the root element in the XML schema. b) all the attributes of this object is listed as the attributes of the root node. c) all the classes which are in association with the focal object are included as a child of the root. d) this process will continue until there is no class in association with the parent node.

Message element identification: the proposed framework does not address a database integration problem but an application layer integration. Therefore, while exchanging messages between two application domains with different databases, "ID" selection for each message element is an issue. Therefore, there is not a possibility of computer generated IDs for each message element. Our approach is to ask domain experts to agree upon a combination of unique set of real world attributes for each entity as their ID. For example an "ID" for a person could be his Social Insurance Number or his Passport Number or a combination of these to distinguish between different message elements in an interaction.

Common message element set mapping: In this step, all the generated common message elements are mapped to extracted target message element set and vise versa using the defined mapping procedure in 5.4. This mappings are performed to transfer information among local standards. Based on the procedure defined in Section 5.4 the source message element set is mapped to the target set and the target set is ready to be harmonized with the static back-bone information model. All the mappings are stored to have transparency among local standards.

Common terminology mapping to local terminology systems: A directory of terminology mappings with a key per concept which is equal to concept codes from the shared terminology system is created. This step, requires deep knowledge of local terminology systems which is provided by terminology experts in involving domains. The terminology mapping is stored in an XML file.

5.3 Dynamic Design Process(HDF):

This process is performed to produce functional specifications, interface specifications, and document receiver's responsibilities. This process may conclude in further refinement of the design information model obtained from the previous process. Design dynamic process is demonstrated in Figure 5.4 including its steps.

The inputs for this section are the design information model and collaboration point analysis model for business analysis of the functional requirements.

The detailed elaboration on steps of this process are as follows:

Define Interaction Triggers : In this step all the required pre-conditions or interaction triggers are specified based on the analysis performed by in collaboration point analysis process.

Further description is as follows: a) based on the design information model, the class which is the object of the interest for the business process is identified. b) based on the interactions in the business process, conditions for information exchange and behavior of the receiving system is defined. c) the conditions which cause information exchange are modeled by UML state charts illustrating the focal object's state change. d) the user-triggered interactions are modeled by activity diagrams.

Document Notification Triggers: using the previous step outcome, the state transitions of "focal" objects which are triggered by system interactions will be recorded. In this step, as the exchange information requirements are clearly revisited and more clarified and the design information model is refined if necessary.

Specify the type of interactions and system roles: interactions are of three types: a) notification triggers which is triggered by business objects state transitions, b) Request/Response which are triggered by end-users only, c) Query/Response which are triggered by end-users only, System roles are categorized it follows: informer, tracker, placer, or fulfiller. All the interactions and their responsible involved roles will be categorized based on the above criteria. These analysis will be recorded as interface specifications.

Document Receiver's responsibilities: Each system role as a receiver has several responsibilities and actions to perform after receiving a message. Therefore, all these responsibilities and actions are defined in each interaction as an attachment (we keep this attachment as a plain text.)

a) all the state transitions for the receiver's behavior is documented in this step. b)

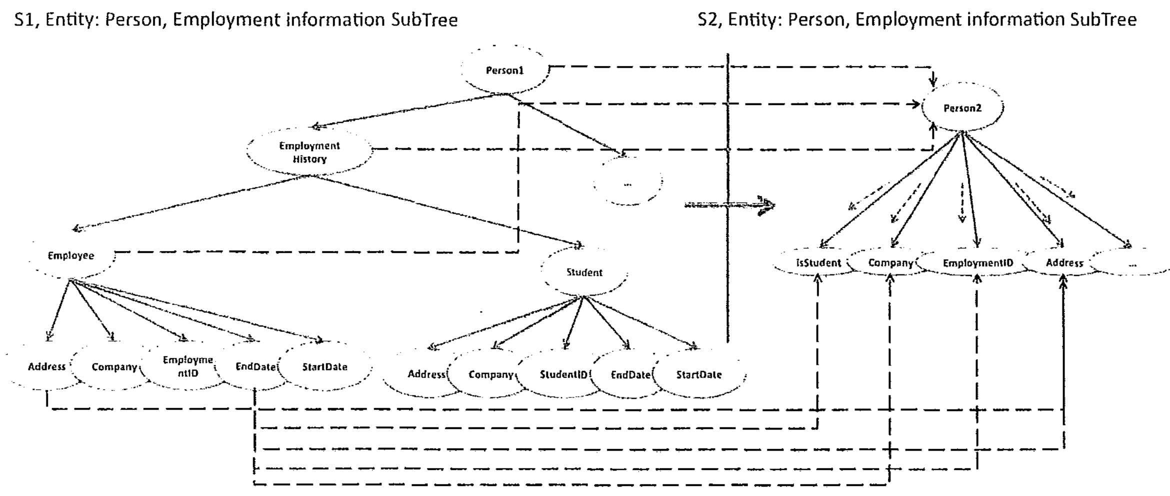

Figure 5.5: Application of proposed mapping algorithm to map two schemas. Following is the list of all pairs in candidates HashTable: {Person2, {Person1}}, {isStudent, {person1.EmploymentHistory.Student.EndDate}, ...

the input parameters and return types are specified in this step. c) as another analysis is performed in this step, the design information model may be changed as well.

5.4 XML-schema Mapping Algorithm

Procedure 1 SemanticXMLSchemaMapping(*source*, *target*)

candidates: HashTable < *key*, *value* >;

key: *node*;

value: list of <*node*>;

SemanticMatchVisit(*target*, *source*);

for all *v* in *target.children* **do**

  SemanticMatchDFS(*v*, *candidates*[*target*]);

end for

A procedure for semantic matching of two XML schemes are developed and used for

Procedure 2 SemanticMatchDFS(*node*, *candidates*)

SemanticMatchVisit(*node*, *candidates*[*node.parent*]);
if *node* is not an atom **then**
  for all v in *node.children* **do**
    SemanticMatchDFS(v, *candidates*[*node*]);
  end for
end if

the case study under review which is matching HL7 v3 common message elements with ACORD massage schemes. The purpose of this algorithm is finding the exact mapping among atoms (leaf nodes)[26] of the XML message element. One of the participating standards which has the longest XML tree and more atoms is the target standard; it implies the target standard has more abstract levels in modeling information which will help in harmonization phase. Message elements of the other standard is mapped to the master XML tree. This is a transit phase so selecting a standard as a master standard does not make a difference.

Source tree and target tree are procedure 1 inputs. A HashTable, which is a mapping between target nodes and a list of source nodes is set as a local variable and named "candidates". In "candidates" each node in the target tree is assigned to a list of nodes of source tree. All the elements in the "candidates" list is in semantic relationship with the target mode which they are mapped to. Starting from root as target and set source as its candidate list. Afterwards, a depth first search is performed on the target tree and in visiting each node the list of candidates for it will be constructed.

In the visit function a target node named *"t"* and its parent's *"candidates"* list is taken as an input. A breadth first search is performed on each node in *"candidates"* list. During the BFS on each c in "candidates", if c in the queue is in semantic relationship with t and is in the same granularity of meaning it will be added to the t candidate list and if not it will be dequeued from the queue. If c has more information than t, its

Procedure 3 SemanticMatchVisit(*node*, *candidates*)

```
for all v in candidates do
  Create a QUEUE Q;
  ENQUEUE(Q, v);
  while Q is not empty do
    for all v1 in Q do
      if v1 is unexplored then
        if v1 is not in semantic relationship with node then
          DEQUEUE(Q,v1);
        else if v1 is in the same level of granularity as node then
          candidates[node].add[v1];
        else if node is an atom and v1 is an atom too then
          use XSLT template match to map related atoms.
        else if node is an atom and no candidate atom is found then
          set a constant default value for node.
        else if v1 is in semantic relationship with node then
          ENQUEUE(Q, v1.children);
        end if
      end if
    end for
  end while
end for
```

children will be enqueued for further verification. If t is an atom, all the atoms which are in semantic relationship with it and generate the exact mappings. In our approach the mappings would be performed by XSLT.

An example is showed in Figure 5.5. S1 and S2 are two domain standards which modeled a person's employment history in two different ways. In S1, all the employment information and the start and end date of each one is modeled, but in S2, all the employment in present is put under person message element. To map S1 person to S2 person, the proposed algorithm is applied and all the candidates for person in S2 is listed in the figure caption. For instance, to map the correct address from person1 to person2, the end date of the employment is important. If the endDate in person1.EmploymentHistory.Employee or person1.EmploymentHistory.student is "present", then the correct address is the address corresponding to this employment. Therefore, EndDate is in semantic relationship with all the fields related to employment information as shown by the dashed arrows.

As the outcome of this phase, we have XML schemes of message elements from each domain, mappings of message schemas. All the message are transformed to Glossary of class diagrams. All the message elements and their attributes are transformed to classes and class properties respectively. Furthermore, nested message elements are represented as associations between classes based on semantic meaning which could be aggregation, composition or other types of associations.

To traverse both trees we made a choice between two methods of tree search, depth first search (DFS) and breadth first search (BFS) on both target and source trees. The purpose of traversing the source tree is to complete the candidates list for each node in the target tree. In the target tree, we are looking for candidates through restricting the node's parent candidates list.

The rational behind selecting DFS for target tree exploration is: by DFS we explore all the atoms in related semantic area of the tree and then we will switch to another area; since finding candidates are done by technical experts, the least switching between semantically related areas in the tree will distract experts more unlikely.

The reason we selected BFS is as follows: to find the candidates we need to explore the tree down to the level of the same semantic granularity as the input node. As the node's parent candidates list is most likely one level upper than the input node, by BFS we can find them sooner and algorithm will stop in a better time constraint in average case.

5.5 Harmonize a class with CIM

This procedure is defined to match a class as an input to CIM back-bone classes. The detailed description of each step is defined as follows:

a) the related subject area will be determined, e.g. the subject area for Policy class from ACORD is "ACTs". b) in each subject area, the class which has the most common attributes and associations with the class is determined. c) to check if the concept of the input class is already in CIM, for the "Act", "Entity" and "Role" related classes "classCode", for "Participation", "ActRelationship" and "RoleLink", "typeCode" sets in each domain specific "CoI" and the "General CoI" should be verified. If the concept exists, the refined classes should be reviewed to investigate if it satisfies the requirements for this information exchange, if not the modify step would be performed. If the concept does not exist, the concept should be created and added to CIM.

- **Create a new concept:** The "classCode" value set for "Act", "Entity" and "Role" or "typeCode" value set for "RoleLink", "Participation" and "ActRelationship" is updated by adding the concept name for the input class. Among all the classes in the selected subject area, the class with the most common attributes and associations is found, and if more attributes or associations are required a new sub-type is created to satisfy the requirements. Afterwards, the data types for each attribute and value sets of them should be specified based on HL7 v3 development and naming conventions.

- **Modify a concept:** If the concept exists in one the checked CoIs, it should be reviewed to meet the interaction data exchange requirements. In the case where the requirements are satisfied, there is no need for change. Otherwise, the data types and value sets are checked and updated accordingly. The required value sets and data types are added to the attribute and new constraints are applied later in CIM. If an attribute is added to the class, a specialization of the existing class is created but with a distinct name. The class is added to the foundation subject area in CIM as well.

Chapter 6

Simulated environment

In this chapter, we provide a simulation environment to achieve message-oriented cross-domain interoperability through artifacts and common message elements produced in the proposed interoperability. We introduce this simulated environment as a prototype for the proposed semantic interoperability road-map. Two architectures are defined in different levels of abstraction, one is a directory architecture and the other one is a web-service based infrastructure for communication among different application domains with different local standards.

The first proposed architecture is aimed at organizing generated artifacts for easier searching and referring. All the artifacts are recorded in structured documents. This architecture which contains a coding framework as well helps to refer to artifacts in a systematic way. The overview of the directory architecture is illustrated in Figure 6.1. All the artifacts which are related to one domain such as list of selected message elements and terminology mappings are stored in a directory which is named as domain's name. The other artifacts which are used between two domains are stored in the directory with domain1-domain2 name. Examples of these artifacts are storyboards, use case diagrams and activity diagrams.

The second architecture is an infrastructure for message-oriented communication. To tackle the complexity of network-centric interoperability, the trend is towards ease of use, vendor/language/platform independency, and in general raising the level of communication abstraction from low-level techniques to provider-independent techniques, and finally to high-level abstractions such as service oriented architecture (SOA). These technologies to a large extent have diminished the problem of interoperability of heterogeneous data among distributed systems. The proposed simulation environment is based on SOA and the technology of choice to implement such an architecture is document-based web-services.

All the messages are represented as XML documents and their schemas are represented in XSD format. Each individual interaction is implemented as a web service and messages are transmitted using SOAP messaging standard. Web services are implemented as document-based web services to obtain asynchronous interoperability which is the most probable interoperability pattern among different domains.

This chapter is structured as follows: in Section 6.1 the artifact repository and coding system is demonstrated. in Section 6.2 the infrastructure of communication is addressed and all the utilized technologies are introduced. The utilized technologies are XML, XML Schema, XPath, XQuery, XSL/XSLT.

6.1 Artifact repository architecture

In this section, the architecture of artifacts storage is discussed. This architecture is illustrated in Figure 6.1. Each artifact is stored as a structured document with the following general attributes: id, code, title, text, effective time, availability time, reasonCode, languageCode, versionNumber, bibliographicDesignationText and some specialized attributes for each artifact. All "id" attributes for artifacts are auto-generated "id"s by an automated serialized id generator tool. This id generation method requires to change in real world applications based on implementation platforms and document repository fa-

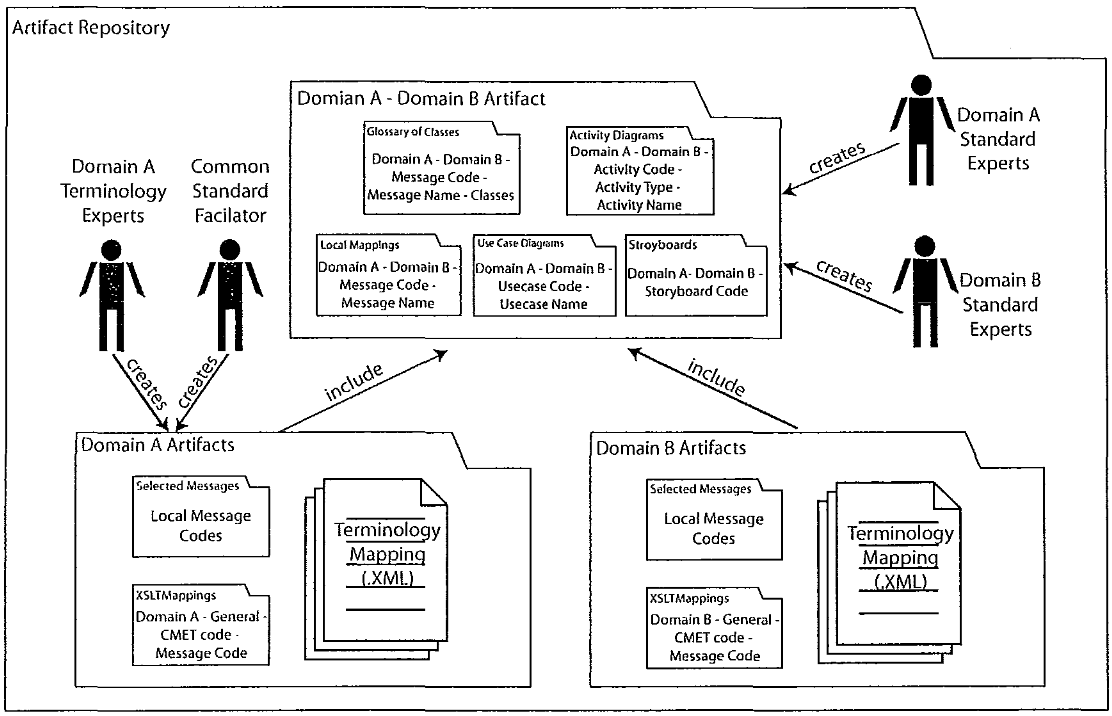

Figure 6.1: Artifact repository architecture. All the artifacts generated in framework processes are stored in a structured repository.

cilitators. All of the involving domain's codes start from 01 and it is a two digit number, e.g. in our coding style healthcare domain code is 01 and insurance domain code is 02.

All of these attributes with their code generation process is discussed below separated by main processes of the framework:

Collaboration Point Analysis: Artifacts which are generated in this process are coded and transformed to structured documents based on the following guidelines.

Storyboards: Storyboards are developed based on communication requirements between two domains (Domain1 and Domain2) who provide the proposal. storyboards are written in plain text. Code attribute for each storyboard is "Domain1Code-Domain2Code-StoryboardCode". Storyboard codes start from 0001 between two domains and they are represented in four digit numbers. Storyboards can not be reused and they are used for one specific collaboration point. These storyboards are maintained in a repository in .txt format (Any other plain text format for this documents are acceptable as well.)

Use case diagrams: use cases are reusable design elements (i.e. they may be utilized in other collaboration points). Use cases are represented in UML 2.0 use case diagrams and converted to images with .jpg extension to be readable on any platform. The code attribute of use cases are Domain1Code-Domain2Code-UsecaseCode-UseCaseTypeCode-UseCaseName. Each use case is annotated with a set of keywords and a description which will be maintained in bibliographicDesignatedText attribute. The set of keywords is used to find the use cases by keyword search for further reuse in other collaboration points.

Interactions: interactions are represented as UML 2.0 activity diagrams. Generated activity diagrams is converted to images with .jpg extension. Interactions are coded the same as use cases: Domain1Code-Domain2Code-InteractionCode-InteractionTypeCode-InteractionName. Each interaction is annotated with a set of keywords and a description which is maintained in bibliographicDesignatedText attribute. The set of keywords is used to find the interactions by keyword search for further reuse in other similar collaboration points.

Message elements extracted from local domain standards: For the purpose of simplicity, a table of each local standard message elements. Each message extracted from each local standard is maintained in a spreadsheet. The local code and name and any required comments are added to it. This is stored in a XLS spreadsheet format. Each message element is checked to indicate that one (or more) mapping(s) of this message element exists. A common-style code is assigned to each local message element as Domain1Code-Domain2Code-MessageCode-MessageName.

Local message element to local message element mappings (XSLT files): two sets of message elements which are selected for a specific interaction, are mapped to each other. This mapping is performed by using Altova XMLSpy 2008-Altova Map Force tool. We load two full version of the message element set included in one root tag from each local domain in the mapping tool and consider of them as source and the other as target. This XML files is shown as trees in this tool and we select to map them by selecting the target and source of the mapping. We apply the proposed procedure to map these two element sets step by step. An XSL file is generated which contains the transformation of these two message sets. Messages are mapped according to one each field of terminology since the terminology mapping is proposed. These mappings are coded as Domain1Code-Domain2Code-sourceMessageCode-targetMessageCode. sourceMessageCode and targetMessageCodes are codes of source and target sets from message domain code eliminited.

Glossary of classes and attributes: the target xml file is transformed to a class diagram according to the guidelines of this step. This class diagram is stored as an image with .jpg extension. Its code is the same as the final message element from previous step and a postfix of classes is added.

6.1.1 Design Harmonization process:

in this step we are supposed to harmonize final glossary of classes with CIM based on the mentioned guidelines in harmonization process and generate a set of common message elements. In our case study as the target message element sets are composed of HL7 v3 common message elements, this step is skipped. This step is considered as a future work when other domains other than healthcare are involved and use the framework.

Design dynamic process: for each interaction produced in each activity diagram from business flow analysis step, interaction triggers and pre-conditions depending on each interaction type will be documented and attached to the interaction.

Common message element to local message element mapping: common message elements Are mapped to the local target message element set are mapped together. This mapping is performed by using Altova XMLSpy 2008-Altova Map Force tool. The XML file is represented in a tree structure. We apply the proposed proce-

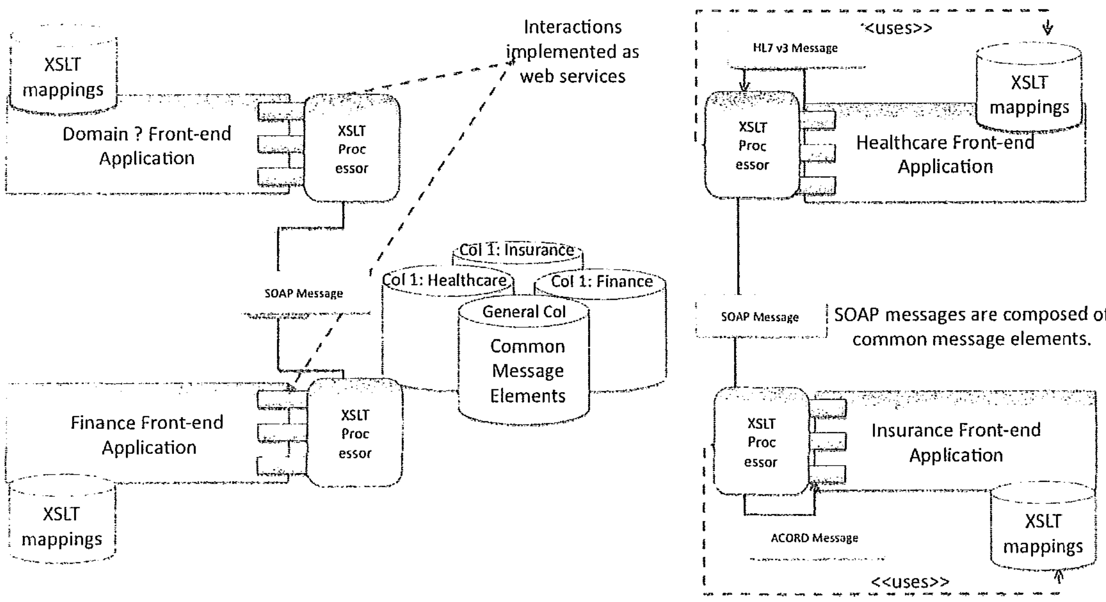

Figure 6.2: Message exchange architecture. Interaction are implemented as web services. All the messages exchanged among different application domain web services are common message elements. These common message elements will be transformed to each local standard message format by invoking an XSLT processor which is embedded in each web service.

dure of mapping step by step. An XSL file is generated which contains the mapping of these two message sets. These mappings are coded as Domain1Code-Domain2Code-CommonMessageElementSetCode-targetMessageCode. sourceMessageCode and targetMessageCode s are the code for the message except for domain codes at the beginning.

Terminology mapping: in this context,we use SNOMED CT terminology system codes for common message element sets. All the value sets in ACORD Life and Annuity documents are mapped to its corresponding code in SNOMED CT and stored in an XML file. The desired SNOMED CT concept and code is found by keyword search.

6.2 Message exchange architecture

In this section, we provide an implementation to use common artifacts provided in the artifact repository. We introduce a simulated environment as a prototype for the proposed semantic interoperability roadmap.

This Section is structured as follows: in Section 6.2.1 XML family of technologies are described. Moreover, an overview of the XSLT processor which is used in this context is addressed. in Section 6.3 basic concepts in document-based web services are discussed and a brief introduction to basic concepts of webservices such as SOAP, UDDI, and WSDL is performed. In Section 6.4, the simulated environment is described and finally and example of applying the information building process through complete semantic interoperability is expressed.

6.2.1 XML family of technologies overview [10]

Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language. By construction, XML documents are conforming SGML documents. XML was developed by an XML Working Group (originally known as the SGML Editorial Review Board) formed under the auspices of the World Wide Web Consortium (W3C) in 1996 [10].

6.2.2 XML Schema [17]

XML schema is intended to define a class of XML documents and the term "instance document" is referred to an XML document which conforms to a, XML schema. Simple and complex types can be defined in XML schemas and be used in document instances. An XML schema document is composed of a schema element and a variety of subelements, most notably element, complexType, and simpleType which determine the appearance of elements and their content in instance documents. Each of the elements in the schema has a prefix xsd: which is associated with the XML Schema namespace through the declaration, xmlns:xsd="http://www.w3.org/2001/XMLSchema", that appears in the schema element. The prefix xsd: is used by convention to denote the XML Schema namespace, although any prefix can be used. The same prefix, and hence the same association, also appears on the names of built-in simple types, e.g. xsd:string. The purpose of the association is to identify the elements and simple types as belonging to the vocabulary of the XML Schema language rather than the vocabulary of the schema author. For the sake of clarity in the text, we just mention the names of elements and simple types (e.g. simpleType), and omit the prefix [17].

6.2.3 XSL/XSLT [13]

A transformation expressed in XSLT describes rules for transforming a source tree into a result tree. The transformation is achieved by associating patterns with templates. A pattern is matched against elements in the source tree. A template is instantiated to create part of the result tree. The result tree is separate from the source tree. The structure of the result tree can be completely different from the structure of the source tree. In constructing the result tree, elements from the source tree can be filtered and reordered, and arbitrary structure can be added.

A transformation expressed in XSLT is called a stylesheet. This is because, in the case when XSLT is transforming into the XSL formatting vocabulary, the transformation functions as a stylesheet. A stylesheet contains a set of template rules. A template rule has two parts: a pattern which is matched against nodes in the source tree and a template which can be instantiated to form part of the result tree. This allows a stylesheet to be applicable to a wide class of documents that have similar source tree structures.

A template is instantiated for a particular source element to create part of the result tree. A template can contain elements that specify literal result element structure. A template can also contain elements from the XSLT namespace that are instructions for creating result tree fragments. When a template is instantiated, each instruction is executed and replaced by the result tree fragment that it creates. Instructions can select and process descendant source elements. Processing a descendant element creates a result tree fragment by finding the applicable template rule and instantiating its template. Note that elements are only processed when they have been selected by the execution of an instruction. The result tree is constructed by finding the template rule for the root node and instantiating its template. In the process of finding the applicable template rule, more than one template rule may have a pattern that matches a given element. However, only one template rule will be applied.

A single template by itself has considerable power: it can create structures of arbitrary complexity; it can pull string values out of arbitrary locations in the source tree; it can generate structures that are repeated according to the occurrence of elements in the source tree. For simple transformations where the structure of the result tree is independent of the structure of the source tree, a stylesheet can often consist of only a single template, which functions as a template for the complete result tree. Transformations on XML documents that represent data are often of this kind. XSLT allows a simplified syntax for such stylesheets. XSLT makes use of the expression language defined by XPath for selecting elements for processing, for conditional processing and for generating text [13].

6.2.4 XPath [14]

XPath is the result of an effort to provide a common syntax and semantics for functionality shared between XSL Transformations and XPointer. The primary purpose of XPath is to address parts of an XML document. In support of this primary purpose, it also provides basic facilities for manipulation of strings, numbers and booleans. XPath uses a compact, non-XML syntax to facilitate use of XPath within URIs and XML attribute values. XPath operates on the abstract, logical structure of an XML document, rather than its surface syntax. XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document.

In addition to its use for addressing, XPath is also designed so that it has a natural subset that can be used for matching (testing whether or not a node matches a pattern); this use of XPath is described in XSLT.

XPath models an XML document as a tree of nodes. There are different types of nodes, including element nodes, attribute nodes and text nodes. XPath defines a way to compute a string-value for each type of node. Some types of nodes also have names. XPath fully supports XML Namespaces. Thus, the name of a node is modeled as a pair consisting of a local part and a possibly null namespace URI; this is called an expanded-name.

The grammar specified in this section applies to the attribute value after XML 1.0 normalization. Within expressions, literal strings are delimited by single or double quotation marks, which are also used to delimit XML attributes. To avoid a quotation mark in an expression being interpreted by the XML processor as terminating the attribute value the quotation mark can be entered as a character reference Alternatively, the expression can use single quotation marks if the XML attribute is delimited with double quotation marks or vice-versa.

One important kind of expression is a location path which is used widely in the communication infrastructure. A location path selects a set of nodes relative to the context node. The result of evaluating an expression that is a location path is the node-set containing the nodes selected by the location path. Location paths can recursively contain expressions that are used to filter sets of nodes. A location path matches the production LocationPath [14].

6.2.5 XSLT processors [41]

An XSLT processor is the software which transforms an XML file to the formatted output which is specified by its XSL transformation templates.

Currently there are three processors that are widely used for XSLT processing because they most closely conform to the XSLT specification:

- **Saxon** Saxon (http://saxon.sourceforge.net/) was written by Michael Kay, the author of XSLT Reference, one of the best books on XSLT. Saxon is a free processor written in Java, so it can be run on any operating system with a modern Java interpreter. Saxon now comes in two flavors: Saxon 6 which handles the XSLT 1.0 standard, and Saxon 8 which handles the newly emerging XSLT 2.0 and other new XML standards.

- **Xalan** Xalan (http://xml.apache.org/xalan-j/index.html) is part of the Apache XML Project. It has versions written in both Java and C++, both of them free. The Java version is described in this book because it is highly portable and easier to set up. Generally Xalan is used with the Xerces XML parser, also available from the Apache XML Project.

- **xsltproc** The xsltproc (http://xmlsoft.org/XSLT/) processor is written in C by Daniel Veillard. It is free, as part of the open source libxml2 library from the Gnome development project. It is considered the fastest of the processors, and is highly conformant to the specification. It is much faster than either of the Java processors. It also processes XIncludes.

Some other processors are used in specific communities, for example microsoft has its own XSLT processor but it is nor open source neither free.

For the purpose of this architecture implementation we use saxon [41].

6.3 Web services

First, we will introduce some basic concepts about web service technology and afterward, we will focus on document-based web services and their implementation approaches.

6.3.1 Web services [8]

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. A web service is defined as below by:

"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."

A Web service is an abstract notion that must be implemented by a concrete agent. The agent is the concrete piece of software or hardware that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided .

6.3.2 WSDL [8]

The mechanics of the message exchange are documented in a Web service description (WSD). The WSD is a machine-processable specification of the Web service's interface, written in WSDL. It defines the message formats, datatypes, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent. It also specifies one or more network locations at which a provider agent can be invoked, and may provide some information about the message exchange pattern that is expected. In essence, the service description represents an agreement governing the mechanics of interacting with that service [8].

6.3.3 SOAP [9]

SOAP Version 1.2 (SOAP) is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

Two major design goals for SOAP are simplicity and extensibility. SOAP attempts to meet these goals by omitting, from the messaging framework, features that are often found in distributed systems. Such features include but are not limited to "reliability", "security", "correlation", "routing", and "Message Exchange Patterns" (MEPs). While it is anticipated that many features will be defined, this specification provides specifics only for two MEPs. Other features are left to be defined as extensions by other specifications [9].

6.3.4 Document-based web services [43]

Java 2 Platform, Enterprise Edition (J2EE) is a standard Java platform for deploying enterprise applications and is specified as a set of required APIs, specifications and policies. The J2EE 1.4 platform provides integrated support for web services, which is one of the many service delivery channels in the platform. The Java API for XML-based RPC (JAX-RPC), a part of J2EE 1.4, is the Java API of choice for developing and using service endpoints based on SOAP that are described using WSDL. Although JAX-RPC and its name are based on the RPC model, it offers features that go beyond basic RPC. It is possible to develop web services that pass complete documents and also document fragments.

Architects and developers are rapidly adopting the use of XML as the data format and a J2EE-based technology stack to develop web services that either expose core business functionality to business partners, or as a mechanism to integrate applications within the enterprise and eliminate vertical application silos.

It is therefore important for them to understand how such document-driven web services can be built using JAX-RPC, the different architectural choices, and the associated tradeoffs.

A web service typically exposes coarse-grained enterprise services that encapsulate some core business and relies on XML-based technologies to do so.

The service consumer can interact with the service in two common patterns. In a document-based interaction, the service consumer interacts with the service using documents that are meant to be processed as complete entities. These documents typically take the form of XML, which is defined by a commonly agreed upon schema between the service provider and service consumer. It is also possible that the document exchanged in such an interaction could be in a format other than XML (such as encrypted files); however, the value of agreeing on a XML schema is to facilitate interoperability.

There are several development strategies for document-based web services based on JAX-RPC technology stack. The decision on formatting style of web services to use in the proposed architecture are as the following:

- **State maintenance:** if the stubs generated by a toolkit cannot maintain state, then document style can be used to pass the contents of an entire transaction as an XML document. The service implementation can then ensure the processing sequence and maintain state in the execution of that sequence.

- **Industry standard schemas:** if the service consumer is only requesting information or persisting information in a pre-defined format, such as those defined by industry standards bodies (eg message-oriented interoperability standards), a document style message makes more sense because it is not constrained by the RPC-oriented encoding.

- **Validate business documents:** with document style, a web service endpoint can use the capabilities of a validating parser and the runtime to perform syntactic validation on business documents against their schema definitions. In order to enforce similar validation with RPC, a message must include an XML document as a string parameter or attachment and implement the validation in the service. This can often lead to invocations that are not detected until the entire structure has been processed. In short, if the service is accepting or returning a complex XML structure, a document style is better suited, since the XML can be validated against the schema prior to calling the service.

- **Performance and memory limitations:** marshalling and un-marshalling parameters to XML in memory can be an intensive process. Typically, the RPC-encoded scheme is the least performing because of the extra processing overhead in encoding the payloads. Also, the SOAP model inherently requires DOM-based processing of the envelope, which can lead to large DOM trees in memory if the XML representation is complex. However, document style services can choose alternate parsing technologies like SAX and StAX to optimize and improve performance, which can be a critical factor for services that handle many simultaneous requests.

- **Interoperability:** there is a natural tendency to expose the programming language object structures through the WSDL when using RPC style and this causes interoperability issues across platforms. To facilitate interoperability, the WS-I Basic Profile limits the use of the encoding (RPC-encoded or document-encoded) and encourages a literal formatting (document-literal or RPC-literal style). Of the two literal styles, some toolkits today like .NET only support document-literal, and if the web service wants to interoperate with service consumers that use such toolkits, document-literal is the natural choice.

Among the above criteria, in the proposed interoperability process, existence of induatry standard schemas, business document validation and interoperability are required. Therefore, we decided to choose document style web services.

Development of document-based web services basically starts with schema definition and WSDL describing document exchange among web end-points. There are some best practice patterns which can be employed in development of web services which are as follows and each of which has its own Pros and Cons.

1. Using XML in the SOAP body.
2. Using String in the SOAP body.
3. Using base64 Encoded or raw bytes in the SOAP body.
4. Using no data binding.
5. Using the xsd:any element in WSDL.
6. Using the xsd:anyType in WSDL.
7. Using an external URI to reference the business document.
8. Using message attachments in the SOAP message.

According to requirements of the implementation process, we chose to use and external URI to reference the business document. The approach to develop a web service URI is explained as the following:

It is possible for the service consumer to send a SOAP message that contains a reference to the business document, and not the actual document itself. The service provider can then use the reference to resolve and obtain the business document in a separate call. In XML Schemas, the anyURI data type is used to represent a Uniform

Resource Identifier, commonly referred to as a URI. This type is mapped to the Java data type java.net.URI (in J2SE 1.4 only) or a String and can be used to convey the location of the business document. The pros and cons are described as it follows which are in line with the project requirements.

Pros:

- The strategy minimizes the payload size since only a reference to the data and not the data itself is contained in the message.
- The data can be dynamically generated if necessary as and when the service provider requests it, or even be cached in proxy servers for optimized processing. This can be useful when dealing with repeated reads that use high latency EIS resources.
- Architectures can be built where the service producer obtains the business document at a later time, thus enabling asynchronous processing. However in such cases, the semantics relating to errors in processing networking need to be carefully predefined.
- Data binding APIs like JAXB can be used directly by the service provider to parse the business documents.
- Can be useful when the same service needs to process multiple documents based on different schemas.

Cons:

- Additional network hops and increased IO introduced in the architecture may increase latency.
- The data may have become stale in the context of the business process by the time the service consumer accesses it.

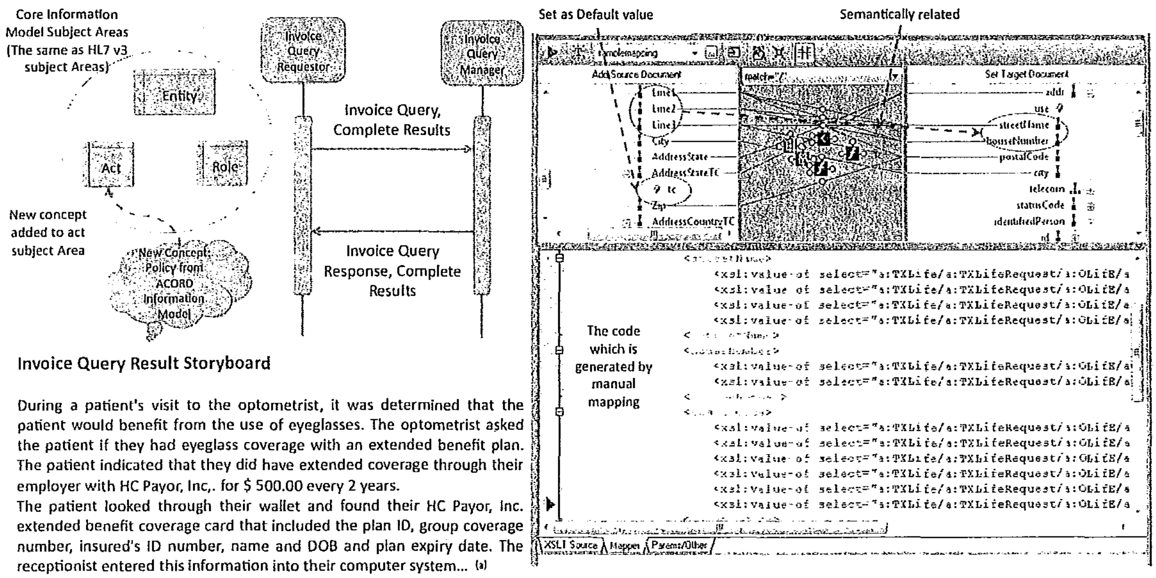

Figure 6.3: Case Study for the proposed cross domain interoperability among two domains healthcare and insurance. Left: storyboard, harmonization process, concept mapping. Right: details of mapping between two XML schema.

- The client needs to have access to a web server or a way to serve up the XML documents when the web service requests them. There also has to be an out-of-band negotiation regarding security and error handling mechanisms for the same.

- Although the XML document is self describing in that it contains a reference to its schema, out-of-band negotiation may be needed to establish this relationship since using the xsd:anyURI essentially obscures this from the WSDL.

- Possible security implications of such an approach need to be considered by architects, since spurious clients can introduce malicious code on the server side.

6.4 Message-oriented interoperability architecture

As a real world case study, we built common message elements between insurance and healthcare domains, according to the proposed information model building framework.

In HL7 v3 ballot [2], storyboards describe the interactions between insurance and healthcare domains. The number of storyboards per topic within the *claims and reimbursement* universal domains are as follows: 6 for eligibility, 6 for authorization, 4 for coverage extension, 12 for pre-determination, 14 for invoice, and 3 for payment. All these storyboards are the input for the use case analysis step in collaboration point analysis phase and 60 interactions are extracted from these storyboards.

In information analysis step in collaboration point analysis process, message elements for each interaction is extracted from HL7 v3 and ACORD. 68 CMETS are extracted from HL7 v3 and mapped to 122 objects in "ACORD Life, Health and Annuity" reference information model based on the procedure proposed using Stylus Studio 2009 Release 2 XML Enterprise Suite as a XML visualization tool. For instance Address in ACORD sample message is in semantic relationship with HL7 v3 addr field in identified person, and further on, Line 1, Line 2 and Line 3 are in semantic relationship with street number and house number in HL7 v3 message. Also as country type code in ACORD is not semantically related to any field of HL7 v3 is should be to the static proper value. These mappings are coded in XSLT through Stylus mapper feature.

Following the collaboration point analysis phase, the collaboration point analysis model is generated composed of all the artifacts showed in Figure 6.1. Moving forward to design harmonization process, all the classes in glossary of classes artifact are mapped into the CIM. This mapping is based on guidelines in design harmonization process phase. As an example, the concept "policy" in insurance which is extracted from ACORD is added to the CIM. First, the subject area is determined which is "ACTs" and the related class in this area is "Act". "Policy" as a concept is added to "classCode" datatype table for "Act" class in CIM and all the other attributes and associations are mapped to the existing CIM which will satisfy the requirements for "policy" concept. Following HL7 v3 refinement process, common message elements are generated. These common message elements are added to related CoI which are mainly General, Healthcare and Insurance CoIs. Further on, design dynamic models are generated for all the interactions. Application roles and business triggers are captured and documented as guide to make the transactions happen between the two domains.

XSLT files are generated from reverse mappings between both ACORD and HL7 v3 message schemas to common message elements and used for the implementation part to address the transparency feature of the framework.

Table 6.1: Comparing common message elements to ACORD message elements-scattering measurement

Scattering Measurement	**One to One** Mappings	**Many to One** Mappings	**One to Many** Mappings
ACORD to HL7 v3	7	12	39
HL7 v3 to ACORD	7	93	22

To measure the complexity of mappings between common message elements and specific domain message elements, we have provided a table which consists of types of mappings among message elements.

There are three types of mappings: i) One to one mapping: when all the attributes of a message element are fully mapped to attributes of one and just one message element in the other message format. The more we have one to one mappings of message elements between two standards, the more the involving standards are similar and the less mappings are complicated. ii) Many to one: when two or more message elements from the source domain standard are mapped to just one message element in the target standard. The more are many to one mappings, the more modeling information in source domain standard is fine-grained. iii) One to many: when the attributes of one message element of the source tree is mapped to attributes scattered in several message elements in target tree. The more are one to many mappings, the more flat is the source standard in modeling information.

In Table 6.4, the number of each mapping type for mappings between common message elements generated in the case study and ACORD message elements are illustrated. As it is shown in the table, ACORD standard is flatter than the common message element structures since we have more one to many mappings in the first row and more many to one mappings in the second row.

Once all common message elements and mappings are ready, we start transmitting them among two front-end prototyped application. We assume one of these two applications uses HL7 v3 as its local standard and the other is an insurance application which uses ACORD as its interoperability standard. As illustrated in Figure 6.2, the infrastructure of cross-domain interoperability is composed of each domain front-end applications, interfaces exposed as document-based web services, repositories of common message elements in each domain CoI and XSLT mappings between common message elements and domain standard message elements.

The local domain applications implement all the interfaces, which are generated in design dynamic process, as document-based web services. Implemented web services are developed under Eclipse 3.4.2 IDE and using wscompile tool to generate stubs and proxies to web services. For the purpose of simplicity, the prototype web services implemented as simple interfaces and no background logic is behind them.

These web services use SOAP messaging to communicate. All the SOAP messages refer to actual message elements which are of common message element types by a URI. These common message elements are then transformed to local message formats to be consumed in local applications which use their local interoperability standard.

We used sax XSLT processor package. the code we embedded in each web service to perform the mapping is as the following:

//instantiate a processor XSLProcessor processor = new XSLProcessor();

//loading the xsl file InputStream xslInput = new FileInputStream(xslFile); XSLStylesheet stylesheet = processor.newXSLStylesheet(xslInput);

//parse the xml input file (which can be referred to as the source tree) DOMParser parser = new DOMParser(); parser.retainCDATASection(false); parser.setPreserveWhitespace(true); parser.parse(xmlFile); XMLDocument xmlInput = parser.getDocument();

XMLDocumentFragment result; result = processor.processXSL(stylesheet, xmlInput);

Once the transformation is performed, the local message is passed to the invoked web service. The result message is transformed to the common message element by the reverse XSLT file. With the proposed infrastructure, transparency among different domain applications are provided.

Chapter 7

Conclusion and future work

In this paper we have proposed a framework to have a common set of message elements among different application domains. Following steps in the proposed framework, common message elements between healthcare and insurance generated. Intermediate documents such as storyboards, use case analysis, process flow, mapping between ACORD messages and HL7 v3 and associated XSLT files, mapping between glossary of classes and Core Information Model in Excel file format, and finally interaction documents from the dynamic design phase is generated.

In all of these steps all the information modeling quality is assured. This framework comparing to existing ones, has both benefits of point to point semantic interoperability and Hub-and-Spoke model. The evolutionary feature of this framework of information model building makes it more practical and useful and ready to use as well.

The proposed framework covers a significant part of interoperability issues among application domains; therefore, it originates considerable new research topics which requires detailed walk through. We list a few of them here:

1. Add a new interoperability standard (e.g. banking standards) to asses the scalability of the proposed framework.

2. Use classification and pattern matching methods to enhance the XML-mapping

algorithm.

3. Specify the role of experts in different areas in the proposed framework.

4. Expand message mapping from ACORD to HL7 v3.

5. Evaluate the proposed framework based on quantities indicators.

6. Utilize a standard coding system to organize artifacts.

Bibliography

[1] Corba. URL = http://www.corba.org/. [Online; accessed 1-August-2009].

[2] Health Level Seven Ballot official website. URL = www.hl7.org/v3ballot/html/welcome/environment/index.htm. [Online; accessed 1-August-2009].

[3] Health Level Seven official website. URL = www.hl7.org. [Online; accessed 1-August-2009].

[4] Open Clinical website. URL = www.openclinical.org. [Online; accessed 1-August-2009].

[5] Oracle HTB official website. URL = www.oracle.com/industries/healthcare/htb.html. [Online; accessed 1-August-2009].

[6] Federal Aviation Administration. Airport mapping standards integration. September 2006.

[7] A. Berler, S. Pavlopoulos, and D. Koutsouris. Design of an interoperability framework in a regional healthcare system. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 2, pages 3093–3096, Sept. 2004.

[8] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. *Web Services Architecture.* W3C, 1 edition, August 2003.

[9] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. *Simple Object Access Protocol (SOAP).* W3C, 1.1 edition, May 2000.

[10] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and Franois Yergeau. *Extensible Markup Language (XML).* W3C, 1.1 (second edition) edition, August 2006.

[11] Lisa Carnahan, Len Gallagher, John Lyons, Charlie McCay Oemig, Lloyd McKenzie Frank Oemig, Jennifer Puyenbroek, and Bas Van Poppel. Refinement, Constraint and Localization, Release 2. URL = http://www.hl7.org/v3ballot/html/infrastructure/, March 2009.

[12] David Chen, Guy Doumeingts, and Franois Vernadat. Architectures for enterprise integration and interoperability: Past, present and future. volume 59, pages 647 – 659, 2008.

[13] James Clark. *XSL Transformations (XSLT).* W3C, 1.0 edition, November 1999.

[14] James Clark and Steve DeRose. *XML Path Language (XPath).* W3C, 1.0 edition, November 1999.

[15] M.J. DiMario. System of systems interoperability types and characteristics in joint command and control. *System of Systems Engineering*, 0:6 pp.–, 2006.

[16] P C Donachy, R H Perrott, T J Harmer, and F Sharkey. Finance sector: Requirements for high assurance within spatial soa based grid infrastructures. volume 0, pages 387–388, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[17] David C. Fallside and Priscilla Walmsley. *XML Schema Part 0.* W3C, primer second edition edition, October 2004.

[18] Jane Grimson, William Grimson, and Wilhelm Hasselbring. The SI challenge in health care. *Commun. ACM*, 43(6):48–55, 2000.

[19] Object Management Group. *Model Driven Message Interoperability (MDMI), OMG Adopted Specification.* OMG, beta 1 edition, March 2008.

[20] Luis Guijarro. Semantic interoperability in egovernment initiatives. *Comput. Stand. Interfaces*, 31(1):174–180, 2009.

[21] HL7. Message development framework, December 1999. version 3.3.

[22] K. Hogg, P. Chilcott, M. Nolan, and B. Srinivasan. An evaluation of web services in the design of a b2b application. In *ACSC '04: Proceedings of the 27th Australasian conference on Computer science*, pages 331–340, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[23] K. Hogg, P. Chilcott, M. Nolan, and B. Srinivasan. An evaluation of web services in the design of a b2b application. In *ACSC '04: Proceedings of the 27th Australasian conference on Computer science*, pages 331–340, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[24] S.H. Hsieh, S.L. Hsieh, Y.C. Weng, T.H. Yang, Feipei Lai, P.H. Cheng, X.O. Ping, M.Y. Jan, J.C. Lin, C.H. Peng, K.H. Huang, L.F. Ko, C.H. Chen, and K.P. Hsu. Middleware based inpatient healthcare information system. In *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, pages 1230–1234, Oct. 2007.

[25] Dean Jin and James R. Cordy. Integrating reverse engineering tools using a service-sharing methodology. In *ICPC '06: Proceedings of the 14th IEEE International*

Conference on Program Comprehension, pages 94–99, Washington, DC, USA, 2006. IEEE Computer Society.

[26] Jaewook Kim, Yun Peng, Boonserm Kulvatunyou, Nenad Ivezik, and Albert Jones. A layered approach to semantic similarity analysis of xml schemas. In *Proceedings of the 2008 IEEE International Conference on Information Reuse and Integration*, July 2008.

[27] Li-Fan Ko, Jen-Chiun Lin, Chi-Huang Chen, Jie-Sheng Chang, Faipei Lai, Kai-Ping Hsu, Tzu-Hsiang Yang, Po-Hsun Cheng, Chia-Chang Wen, Jun-Lian Chen, and Siao-Lin Hsieh. Hl7 middleware framework for healthcare information system. In *e-Health Networking, Applications and Services, 2006. HEALTHCOM 2006. 8th International Conference on*, pages 152–156, Aug. 2006.

[28] Grace A. Lewis, Edwin Morris, Soumya Simanta, and Lutz Wrage. Why standards are not enough to guarantee end-to-end interoperability. In *ICCBSS '08: Proceedings of the Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008)*, pages 164–173, Washington, DC, USA, 2008. IEEE Computer Society.

[29] J. A. Mykkänen and M. P. Tuomainen. An evaluation and selection framework for interoperability standards. *Inf. Softw. Technol.*, 50(3):176–197, 2008.

[30] Katrine Jokinen and Jukka Borgman and Reijo Sulonen. Common data model for design document exchange in business-to-business networks. In *HICSS '05*, page 95.1, Washington, DC, USA, 2005. IEEE Computer Society.

[31] Marijn Janssen and Hans J. (Jochen) Scholl. Interoperability for electronic governance. In *ICEGOV '07*, pages 45–48, New York, NY, USA, 2007. ACM.

[32] Hamid R. Motahari Nezhad, Boualem Benatallah, Fabio Casati, and Farouk Toumani. Web services interoperability specifications. *Computer*, 39(5):24–32, 2006.

[33] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. A software engineering approach to ontology building. *Information Systems*, 34(2):258 – 275, 2009.

[34] College of American Pathologists. SNOMED clinical terms user guide, January 2007.

[35] France Pr. Jean-Pierre BOUREY, Ecole Centrale de Lille. Model driven interoperability and service-oriented architecture. In *IMS Seminar, Zurich*, 2007.

[36] Schahram Dustdar Rainer Anzbock. *Semi-automatic generation of Web services and BPEL processes - A Model-Driven approach*, volume 3649/2005 of *Lecture Notes in Computer Science*, chapter Business Process Management, pages 64–79. Springer Berlin / Heidelberg, 2005.

[37] Rachel L. Richesson and Jeffrey Krischer. Data standards in clinical research: Gaps, overlaps, challenges and future directions. *Journal of the American Medical Informatics Association*, 14(6):687 – 696, 2007.

[38] Mei Y. Selvage, Dan Wolfson, Bob Zurek, and Ed Kahan. Achieve semantic interoperability in a SOA. In *Patterns and best practices*. IBM Inc., 2006.

[39] Health Level Seven. *Healthcare Development Framework (HDF), Methodology Specification*. Health Level Seven, 1.2 edition, April 2008.

[40] Shweta Shetty, Steven Nordstrom, Shikha Ahuja, Di Yao, Ted Bapty, and Sandeep Neema. Systems integration of large scale autonomic systems using multiple domain specific modeling languages. volume 0, pages 481–489, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[41] Bob Stayton. *DocBook XSL: The Complete Guide*. Sagehill Enterprises, 4 edition, September 2007.

[42] The ACORD Life Team. *Life and Annuity and Health Standards Program.* Association for Cooperative Operations Research and Development, 2.19.00 edition, June 2008.

[43] Sameer Tyagi. Patterns and strategies for building document-based web services. *Sun Developer Network (SDN)*, 1, September 2004.

[44] Weiping Wang, Mingming Wang, and Shijun Zhu. Healthcare information system integration: A service oriented approach. In *Services Systems and Services Management*, volume 2, pages 1475–1480, June 2005.

[45] T. H. Yang, P. H. Cheng, C. H. Yang, F. Lai, C. L. Chen, H. H. Lee, K. P. Hsu, C. H. Chen, C. T. Tan, and Y. S. Sun. A scalable multi-tier architecture for the national taiwan university hospital information system based on hl7 standard. In *CBMS '06: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 99–104, Washington, DC, USA, 2006. IEEE Computer Society.

[46] Huimin Zhao and Sudha Ram. Entity matching across heterogeneous data sources: An approach based on constrained cascade generalization. *Data and Knowledge Engineering*, 66(3):368 – 381, 2008.

[47] Ying Zou and Kostas Kontogiannis. Migrating and specifying services for web integration. In *working notes of the International Workshop on Engineering Distributed Objects*, pages 2–3, Univ. California Davis, June 2000. Springer-Verlag.

[48] Ying Zou and Kostas Kontogiannis. Towards a web-centric legacy system migration framework. In *Proceedings of IEEE Workshop on Network Centric Computing*, Toronto, ON, May 2001.