Interoperability of Data and Mined Knowledge in Clinical Decision Support Systems

Interoperability of Data and Mined Knowledge in Clinical Decision Support Systems

by

Reza Sherafat Kazemzadeh

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of
Master of Applied Science

McMaster University

Master of Applied Science(2006)   McMaster University
(Software Engineering)   Hamilton, Ontario

TITLE: Interoperability of Data and Mined Knowledge in
Clinical Decision Support Systems
AUTHOR: Reza Sherafat Kazemzadeh, B.Sc. (Sharif University of Technology)
SUPERVISOR: Dr. Kamran Sartipi
NUMBER OF PAGES: xiv, 136

Abstract

The constantly changing and dynamic nature of medical knowledge has proven to be challenging for healthcare professionals. Due to reliance on human knowledge the practice of medicine in many cases is subject to errors that endanger patients' health and cause substantial financial loss to both public and governmental health sectors. Computer based clinical guidelines have been developed to help healthcare professionals in practicing medicine. Currently, the decision making steps within most guideline modeling languages are limited to the evaluation of basic logic expressions. On the other hand, data mining analyses aim at building descriptive or predictive mining models that contain valuable knowledge; and researchers in this field have been active to apply data mining techniques on health data. However, this type of knowledge can not be represented using the current guideline specification standards.

In this thesis, we focus is on encoding, sharing and finally using the results obtained from a data mining study in the context of clinical care and in particular at the point of care. For this purpose, a knowledge management framework is proposed that addresses the issues of data and knowledge interoperability. Standards are adopted to represent both data and data mining results in an interoperable manner; and then the incorporation of data mining results into guideline-based Clinical Decision Support Systems is elaborated. A prototype tool has been developed as a part of this thesis that serves as the proof of concept which provides an environment for clinical guideline authoring and execution. Finally three real-world clinical case studies are presented.

Dedication

To my beloved family, Rasoul, Roudabeh, Rosa, and Roya.

Acknowledgements

I would like to take this opportunity to thank all those who helped me to accomplish this research. I would like to express my sincere thanks and appreciation to Dr. Kamran Sartipi for his guidance and continues support. We spent countless hours, discussing different aspects of my research and he was a major source of inspiration and determination along the project. Also, special thanks and gratitude to Professor Norm Archer who contributed his time and experience in this field.

Also, special gratitude to Professor Brian Haynes and Dr. Ann Mckibbon who provided me their vast knowledge of healthcare decision support systems and constructive critique. I would also like to thank the examiner committee members, Dr. Doug Down and Dr. Frantisek Franek for their help and time that they put on reviewing this thesis and providing me with constructive suggestions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Due to the paramount importance of the quality of public healthcare services, these services represent a major portion of the government spending in most countries and usually are considered as a significant measure of each country's quality of life. In Canada the Provincial Government of Ontario invested a total of $28.1 billion in healthcare services in 2003-4 [38]; and the Canadian Institute for Health Information (CIHI) speculates that the total healthcare spending throughout Canada reached as high as $142 billion in 2005, representing a 7.7% increase from the year 2004 [19].

However, the high volume of spending in healthcare does not necessarily translate to perfect and error-free health services. While today's healthcare professionals are overwhelmed with information, preventable medical errors are estimated to be the main cause of 44,000-98,000 deaths [14] and loss of up to $29 billion annually in the United States alone [41]. A part of this problem lies in the inherent complexity and the dynamism of today's medical knowledge which is changing fast and constantly on a daily basis.

While only a hundred years ago, it could be claimed that a single medical person can potentially master all the medical knowledge of that era, there is no doubt nowadays that human memory is not capable of storing, organizing and effectively using this staggering amount of knowledge. Advancements in the information management and computer industries on the other hand, have provided some hopes that by adoption of computer-based decision support systems practitioners can tackle the problem of how to access and apply medical knowledge.

Clinical Decision Support Systems (CDSS) are computer applications that assist practitioners and healthcare service providers in decision making by providing access to electronically stored medical knowledge [16]. The medical knowledge might be represented in a variety of formats that are computer-interpretable. The decision is tailored specifically for the patient who is subject to the decision making practice and may be part of a patient diagnosis, treatment or long-term care. Evaluation studies show that computerized Clinical Decision Support Systems are helpful and have positive effects such that the healthcare professionals often report a high degree of satisfaction [26]. It should be noted that any improvement in the healthcare services sector will result in savings that benefit all the stakeholders, from patients to healthcare providers and government agencies both financially and quality of care.

A Clinical Decision Support System stores its operational decision making logic in a knowledge repository which we refer to as the knowledge-base. The system retrieves the patient data either directly from the user (i.e., healthcare professional interacting through the Graphical User Interface) or indirectly from a data source, i.e., the hospital's Electronic Medical Record (EMR) systems. An interpretation engine then applies the knowledge in the knowledge-base for each particular patient and provides the results back to the user.

Data interoperability

Currently, there are several barriers in the widespread implementation and application of healthcare information systems and Clinical Decision Support Systems are not exempt either. It is often the case that patient data is scattered among many healthcare institutions and hence data availability is usually the first concern of healthcare information systems developers. This necessitates effective systems collaboration in the healthcare settings to provide the required data pieces residing at a location to the interested users. While it is relatively easy to have all participating systems connected through a network, the issues of interoperability arise due to the heterogeneity of these systems.

Since each healthcare information system serves a particular purpose, there might be huge variations in the internal data models, and the formats used for storage of the patient data. On the other hand there is a general agreement that different data storage formats are inevitable; since a totally unified data modeling approach results in inefficient storage and is also impractical due to large volumes of patient data that already exist in legacy health data repositories.

Differences in internal data representations may cause problems when data items in one system cross to the other. We refer to the ability of information systems to correctly interpret the data from another system as data interoperability. Data interoperability is two fold. At one hand, the communicating systems should both use the same set of vocabulary terms and data types; this is called syntactic data interoperability. On the other hand, the interacting systems should interpret the terms with the same semantic. This is referred to as semantic data interoperability and is normally much harder to achieve.

The data interoperability problem has been subject to much debate and research in recent years and several standard vocabulary sets and data models have been developed by different health organizations. The most famous well-known attempts are Health Level 7 (HL7) Reference Information Model, and Clinical Document Architecture (CDA), with the aim to provide semantic data interoperability. These standards rely on standard terminology systems to provide syntactic data interoperability.

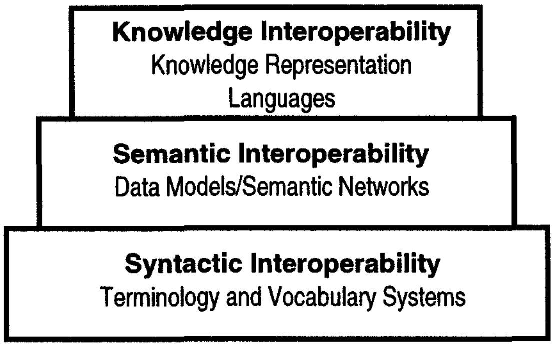
Knowledge Interoperability
Knowledge Representation Languages
Semantic Interoperability
Data Models/Semantic Networks
Syntactic Interoperability
Terminology and Vocabulary Systems

Figure 1.1: Different layers of interoperability.

Knowledge interoperability

In the case of decision support systems, the knowledge interoperability issues also arise, since these systems have to use and interpret the clinical knowledge that may have been represented in different formats. The knowledge repositories that may have been developed by different groups using different methodologies are valuable assets that should be made available to other interested users for application. We refer to this ability to incorporate and use the knowledge produced by different stakeholders as *knowledge interoperability.*

Since knowledge is applied on patient data, data interoperability is a necessary requirement for decision support systems to have knowledge interoperability. Fig 1.1 represents different layers of interoperability that are required for a Clinical Decision Support System to be easily deployed at different institutions. Each lower layer is required for the higher level to be achieved.

Applications of data mining techniques in healthcare produce valuable knowledge. Currently the results of data mining studies would either end up in medical literature, where these results are presented mostly in simple formats and graph diagrams, or be implemented locally as a stand alone software application that is specifically developed for each particular study. Since a lot of effort is usually put into the process of extracting mined knowledge from healthcare data repositories, it is desirable that these results be available to those institutions that are interested in accessing and using the mining results. Unfortunately, porting an application to a new environment is not an easy task and the interoperability problem of heterogeneous software systems is often an issue that is hard to overcome.

To tackle this problem, this thesis adopts standards in the fields of healthcare data modeling and data mining, and extends a clinical decision making standard to achieve the required portability. Data models, data mining models, and clinical guideline models are defined using these specifications that provide a common language for different parties to publish clinical guidelines that use data mining results for decision making. Figure 1.2 illustrates a clinical decision support system that is capable of interpreting data and data mining results as the source of knowledge. These results are extracted in a data mining operation carried out on healthcare data sources. The extracted knowledge is stored in the knowledge-base in the form of PMML (Predictive Model Markup Langauge) models. Patient data is also accessed from healthcare databases in the form of CDA documents. As shown in the figure 1.2, clinical best practice guideline models within the Clinical Decision Support System invoke the logic modules to retrieve both data and knowledge, interpret the data and knowledge, and finally provide the results to the user.

In the rest of this thesis, the term *knowledge* will refer to the results of a data mining operation. This knowledge is extracted in a knowledge discovery process, and has undergone careful evaluation and inspection by expert medical researchers to ensure correctness.

1.1 Motivations and problem statement

This section reviews some of the most important issues in the healthcare industry that serve as motivations for the contributions of this thesis.

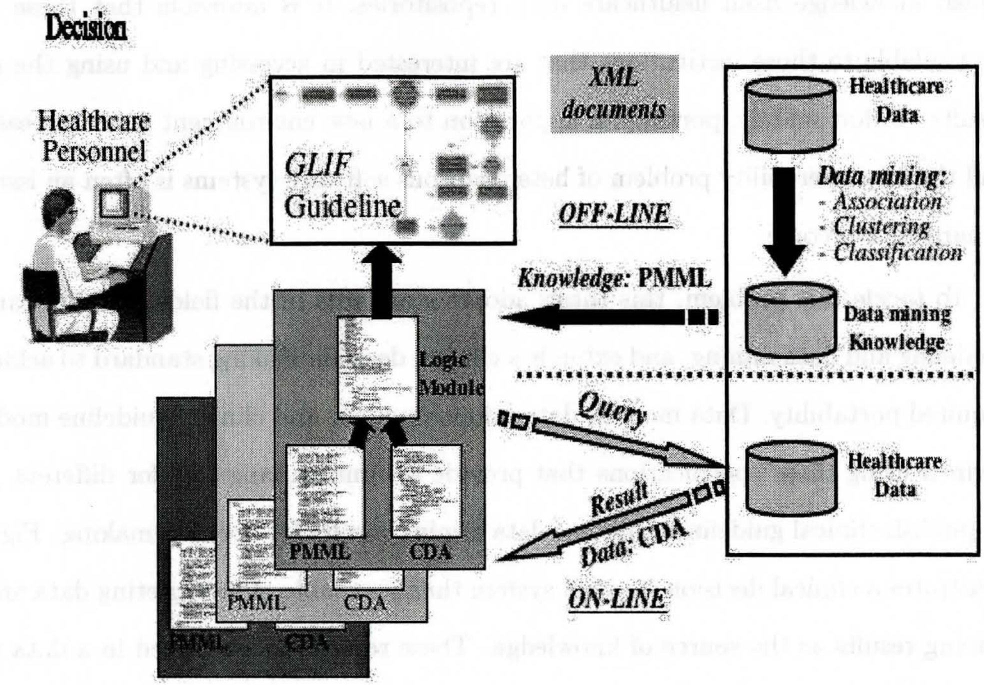

Figure 1.2: A guideline-based Clinical Decision Support System in the heterogeneous healthcare environment that uses mined knowledge for decision making.

- While healthcare research produces viable knowledge, the results are often used in an environment confined to the local provider that provided the test bed for the research. The final systems that are developed usually are in the form of stand alone applications that are dependant on the features of the local information system [37].

- Data mining techniques are also applied to healthcare data and there is no specific approach or methodology for seamless integration of these results in clinical decision support systems. This is mainly because, the data mining models often have a complex structure and are also hard to interpret using the conventional and commonly used logical expressions.

- The healthcare industry suffers extremely from the lack of standardization. Different systems have been developed in an ad-hoc manner that are often hard to be integrated to the already existing and legacy systems. Moreover, after deployment at a site, future systems integrations will also face the same problem.

- The healthcare information systems are deployed in a heterogeneous and distributed environment which often requires systems communication crossing organizations' boundaries. In this environment, the syntactic and semantic data and knowledge interoperability issues are inevitable and very critical.

Based on the above observations we define the research problem in this thesis as:

> *Devising methodologies, techniques, and tools to streamline the dissemination and application of data and mined knowledge for clinical decision making in the distributed and heterogeneous clinical settings.*

1.2 Scope of the problem

The scope of the research in this thesis extends to the data and knowledge interoperability among heterogeneous healthcare systems, and the use of data and mined knowledge within the context of Clinical Decision Support Systems. We try to propose a solution for knowledge interoperability between sources of knowledge and their users. However, we do not directly address the process of extracting the knowledge from healthcare data, the acceptance degree of the results of knowledge extraction, or the method that credible clinical best practice guidelines are developed and validated.

1.3 Proposed approach

In this thesis we address the problem of incorporating the results from data mining analyses into Clinical Decision Support Systems for use at the point of care. The proposed approach relies heavily on adoption of standards to encode healthcare data and knowledge. Eventually we address both the data and knowledge interoperability problems that are key factors in success and adaptation of decision support systems by healthcare stakeholders.

Moreover, to enable data mining supported clinical decision making, we adopt a simplified version of a flow-oriented guideline modeling specification and extend it. The medical best practices are represented in the flow graph as a sequence of patient states, actions, and decision points. To provide interoperable access of the required patient data, standard healthcare data models and data representation standards are adopted and tailored for our particular application requirements. The data mining knowledge is also extracted in a data mining process and the results are encoded and stored in a standard format to achieve knowledge interoperability. At different steps in the flow graph of the clinical guideline the data and mined knowledge are accessed from the data- and knowledge-base and interpreted.

1.4 Contributions

The contributions of this thesis are as follows:

1. Proposing a novel framework that supports interoperable data and (mined) knowledge dissemination in the context of Clinical Decision Support Systems.

2. Extending a simplified version of a current guideline standard, i.e., GLIF3 to enable decision making based on data mining results; while facilitating interoperable data and knowledge access from heterogeneous sources.

3. Enabling GLIF3 models to access data items from CDA documents in an interoperable manner.

4. Developing a prototype tool for clinical guideline modeling and execution that supports data mining-based decision making and uses the extended version of GLIF3.

5. Applying the proposed approach to knowledge dissemination on a real-world clinical data mining case study from the literature.

1.5 Thesis overview

The remaining chapters of this thesis are organized as follows:

Chapter 2: provides an overview of the related work in the area of healthcare decision support systems research. Some of the most recent CDSS projects that are related to the approach in this thesis are analyzed.

Chapter 3: proposes a novel framework for data and knowledge interoperability that is used to incorporate mined knowledge into guideline-based Clinical Decision Support Systems.

Chapter 4: reviews the knowledge discovery process in general, and the healthcare data mining research, in particular. Different types of data mining techniques are described, and examples of application of data mining techniques on healthcare data are provided. A case study that is chosen from the literature is described which will be used through the subsequent chapters to describe the approach of this thesis.

Chapter 5: describes the approach for data and knowledge interoperability in healthcare. First, the chapter focuses on healthcare data modeling standards, vocabularies and terminology systems, and how they collectively provide the necessary semantic and syntactic data interoperability foundations that our proposed framework is based on. The chapter continues with describing our approach in knowledge interoperability.

Chapter 6: provides a definition for Clinical Decision Support Systems that is adopted by this thesis, and reviews some other definitions by other sources. A simplified version of the GLIF3 standard that is extended as part of the contributions of this thesis to incorporate mined knowledge is described. This chapter further describes the guideline modeling and execution environment that was implemented and is capable of interpreting the extended version of GLIF3.

Chapter 7: provides a short conclusion and sets the paths for future works.

Appendix A: describes the application of the proposed framework for interoperability of data and knowledge on a case study. Different documents for encoding sample data, and the mined knowledge are presented in full.

Chapter 2

Related work

In this chapter, we provide a short review of the related work for making healthcare data and knowledge (clinical best practices) available for use in a computer interpretable manner. Some of the Clinical Decision Support Systems that have been developed are described and their approach for representing the clinical decision making logic is discussed[1].

2.1 EGADSS

Evidence-based Guidelines And Decision Support System (EGADSS) [23] is a stand alone application that assists the practitioners at the point of care with automatically generated alerts and reminders. The medical knowledge in this system is encoded in separate modules called Medical Logic Module (MLM). Each module contains the decision making logic for making a single decision, e.g., re-do a test or vaccination after a certain period of time. Running a module may trigger an action. This action normally generates a message that is added into a results document. The final result document contains the messages that were generated by running all MLMs that reside in the system's knowledge

[1] *e*-MS is not a Clinical Decision Support System. We have included it as a good example of a healthcare data interoperability project.

repository and putting the messages together.

To support data interoperability, the system relies on healthcare data modeling standards (Health Level-7 Reference Information Model) [31]; and the Clinical Document Architecture (CDA) standard [30] has been adopted for this purpose. All requests that are made to the system have their data encoded in XML-based documents according to the CDA specification and hence are accessible and interpretable by all parties that are able to interpret the corresponding CDA documents. This is significant, since as a result there will be no restriction for an organization's internal data representation and storage format, as long as the external data interchange format is CDA compliant. Furthermore, the EGADSS results documents also comply with the CDA specification to provide data interoperability of the decision making results.

For encoding the clinical logic in MLMs, EGADSS adopts the Arden Syntax standard [9]. Arden Syntax specifies a syntax and an expressions language to define $if-then$ rules. Relevant patient data items are accessed by each rule and if a rule concludes as *true* then the action specified in the module is executed. The action normally generates a message that complies to the CDA results document. Rules that conclude as `false` are simply ignored.

EGADSS has been developed with this fact in mind that healthcare systems are usually deployed in a distributed environment and remote and location transparent access to the system is crucial. Access to the system is provided through the client-server model. The interpretation engine and the knowledge repository all reside on a server and the decision making service is exposed to the outside world through a well defined interface. The server acts in a passive manner by waiting for requests from clients to arrive. The requests are in the form of CDA documents containing patient data in XML structured format. This data normally comes from an Electronic Medical Records (EMR) system.

After dispatching the request to the server, the client waits for a response. The response which contains clinical recommendations is generated after consulting with the knowledge repository (constituting of a set of MLMs) and concatenating the outputs. At the client side, the client parses the CDA document and retrieves the results.

The EGADSS approach to represent knowledge is different from the knowledge representation format that we have used. EGADSS adopts Arden Syntax to define the decision making logic as a set of rules that are encoded in MLMs. This approach is not able to use mined knowledge. However, the data interoperability approach using CDA is similar to the approach taken in this thesis.

2.2 HELEN

HELEN [44] is a project which mainly aims to study management and the adaptation process of clinical practice guidelines, and is in fact the first attempt to address this issue. Adaptation of guidelines is a necessary step for achieving widespread use, and the designers of HELEN have presented a framework for this purpose [28, 29]. The outcomes of the project are a representation ontology for clinical best practice guideline modeling, and tools for authoring and execution of guidelines.

HELEN is divided into three parts:

1. *The authoring environment* which uses authoring tools to model (author) the guidelines. The guideline models are then encoded in HELEN XML Format and can be stored for later access.

2. *The server environment* within which the execution engine runs. The execution engine receives the authored clinical guidelines and executes the algorithms encoded in the guidelines.

3. *The client environment* which consists of a *Guideline Viewer* application for the users to access the content encoded in a guideline. The guideline viewer communicates with the guideline execution engine and presents the results for the user. The

current guideline viewer accesses the servers through RMI or HTTP messages.

Currently, HELEN guidelines representing the Apnoea Bradycardia Syndrome are developed by the department of Neonatology in the University of Heidelberg [44]. The evaluation part of the project is still underway and the results are not yet available. HELEN uses a modeling language that is section HELEN-cdss. This modeling language is similar to the GLIF3 language that we used for encoding clinical best practice guidelines. In this research, we can use the data mining results for decision making which is in contrast to the simple logical expressions used in HELEN.

2.2.1 HELEN guideline modeling ontology

HELEN guidelines are defined according to a formal specification that is represented as an ontology. The most current version of the specification is now available as a *Protégé* ontology which defines different elements in the guidelines and specifies their relations. Guideline instances that are built according to the specification can be exported and stored as XML documents and be executed (i.e., interpreted) in an execution engine.

The ontology defines a collection of classes (concepts) that are used for modeling. There are five main concepts in the highest level, namely *Diagram_Entity, HELEN_pragmatics, HELEN_Knowledge_Module, HELEN_Adaption, HELEN_Modules* and *HELEN_Guideline.* Table 2.1 summarizes the purpose of the high level classes in HELEN.

Since the HELEN project aimed to study the adaptation process of guidelines, the designers of the ontology classes paid special attention to this issue. For this purpose, the *HELEN_Adaption* class provides the guideline documentation and data variables and constants that are used both locally and globally within a guideline instance. For a successful deployment, the data items of the participating healthcare institutions should be mapped to these variables and constant data items.

HELEN guideline instances capture the clinical knowledge in a set of *knowledge modules.* The knowledge modules can either be in a text, graphic or algorithm diagram

Class name	Class description
HELEN_pragmatics	Contains information about a guideline's implementation, development, clearing and verification
HELEN_Knowledge_Module	Contains descriptions of a HELEN guideline's knowledge modules (text, graphic and algorithm diagram)
HELEN_Adaption	Describes the elements for local adaption of the guideline. This information includes documentation for the guideline and other information that is needed for deploying the guideline in a site.
HELEN_Modules	Contains the modules of a guideline instance.
HELEN_Guideline	Represents a HELEN guideline and contains the knowledge modules that it defines.

Table 2.1: Summary of high-level concepts in HELEN

format. The most sophisticated form is the algorithm diagram format. The algorithm diagram specifies a flowchart-like diagram which specifies the flow of events in a clinical guideline. The steps in the flow represent different actions, decisions or states. The execution engine reads through these states starting from the *HELEN_Start_Step* and guides the user accordingly. Table 2.2 summarizes different constructs of HELEN algorithm diagrams.

2.3 COMPETE

Computerization Of Medical Practices for the Enhancement of Therapeutic Effectiveness (COMPETE) [12] is a Canadian project that intends to bring computer-based decision support facilities for managing diabetes, hypertension, cholesterol, previous heart stroke, and chronic diseases patients.

Class name	Class description
HELEN_Diagnose	Specifies a diagnosis or result.
HELEN_Patient_State	Specifies a special patient or guideline state.
HELEN_Start_Step	Specifies an initial step within the guideline flow. The execution of the guideline starts from this step.
HELEN_Decision	Models a decision in a guideline that changes the direction of flow. There are currently two different types of decision step: One that is based on user choice, or one that is the result of an automatic evaluation.
HELEN_Action	Recommends an action to be taken. Different actions that are currently supported are: Physical examination or analysis, laboratory or technical examination, and therapy or prescription.
HELEN_Control	Represents different control elements in HELEN guidelines, e.g., loops and subtasks.
HELEN_Message	Explicitly specifies a user/user or an execution engine/user communication.
Set_Variable	Specifies the values or expressions to initialize or modify the guideline variables.

Table 2.2: Summary of high level guideline diagram entities in HELEN

After completion of COMPETE I, which mainly focused on implementation of Electronic Medical Records (EMR) for computerization of clinicians' offices, the COMPETE group turned to Clinical Decision Support Systems. During COMPETE II, a diabetes tracker tool was developed to manage the chronic disease and diabetes patients with telephone based reminders. COMPETE III, has continued and extended the scope to more diseases including vascular risk patients, diabetes and cholesterol. COMPETE III is currently being evaluated in a randomized controlled trial.

The designers of COMPETE III have selected 17 variables that are monitored and used in the clinical best practice guideline. The knowledge-base of COMPETE III is a set of guidelines each represented using a function table. Each row in the tables specifies conditional statements in the form of logical expressions. These rows represent possible situations that can trigger an action. The row also contains two messages to be delivered to the patient and the practitioner when the conditions hold. The designers have paid particular attention in the content of the messages to be accurate and simple, and to prevent the user to become overwhelmed by many possible recommendations. As a result, the conditions are so fine grained that the corresponding message contains *only* one possibility. Also, the rules encode a coloring scheme which is used to convey particular meanings to the user. Three colors of red, amber, and green respectively represent whether there is a need for urgent, low, or no attention.

Simplicity and understanability of the tables are considered as the strength of this approach, however the scope of knowledge that can be represented and managed in this approach is limited. Function tables are represented in the Microsoft Excel Format. The guidelines are hence fully human readable and the execution engine will directly read the content of the tables from the file. This eliminates the need for a programmer's intervention in the process of deployment. On the other hand, at the execution time the patient data is the input to all the guidelines in the repository. This has in practice, caused a substantial extensibility problem as the performance of the system is low and

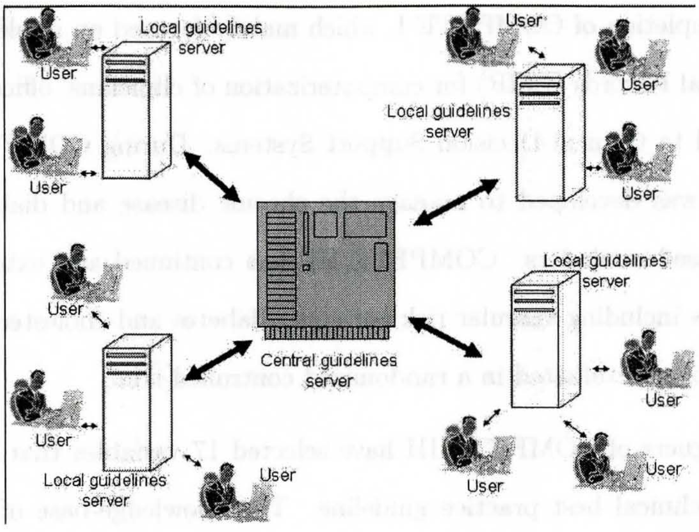

Figure 2.1: The COMPETE III deployment environment.

there is no way to narrow down the guidelines collection.

The implemented version of COMPETE is accessible through the web using standard web browsers that support secure connections. After the data is input by the user or from an Electronic Medical Records (EMR) system, the data is sent to the server in XML format. This message is structured using a Core Data Set specified by an XML schema. At the server, the tables in the knowledge-base are executed and the results are sent back to the user. As mentioned earlier, the scalability of the knowledge-base has been an issue. To overcome this problem, local servers have been deployed to distribute the load from the main server. Local servers will then have to periodically connect to the main server to retrieve new or updated guidelines. This is illustrated in Figure 2.1. In contrast to function table-based representation of clinical knowledge, our approach focuses on the knowledge that is extracted in a data mining operation and encoded as PMML files that are accessed in a flow chart based clinical guideline modeling language. This type of knowledge is not readily representable in COMPETE's function tables.

2.4 ASGAARD project

ASGAARD [6] is another Clinical Decision Support System that uses Asbru as its clinical best practice guideline representation language. Abstu is used to define time-oriented, intention-based skeletal plans. ASGAARD supports continual planning which is a close coupling of planning and execution. This is mainly because of the dynamic nature of the real world, and also to enable the system to cope with exceptional cases. For exceptional cases during execution, an alternative plan is used, or if there is no such alternatives a replanning becomes necessary.

The Asbru guideline representation language is very complex and practitioners find both the language and its concepts hard to grasp. For this reason, several different tools have been developed to support guideline authoring and viewing (at different levels, i.e. topological and temporal), as well as for interpretation and execution. At design time, the authors design the clinical best practice guideline by specifying the conditions, action, intended plan, and intended patient states. The guidelines are stored as XML files and the Asbru interpreter will have them as input. At execution time, the users, i.e. healthcare personnel, apply the guideline by performing the actions that are specified. The current implementations of the execution environment are applications that run on the client machine.

ASGAARD's runtime module is divided into three parts. The first part is the data abstraction module which collects user input or sensor data and performs required transformations to make them suitable for the next module. The second module, the monitoring unit, receives the outputs of the data abstraction module and stores them in a list of Observed Parameter Propositions (OPP). The monitoring unit receives the Monitored Parameter Propositions (MPP) from the execution module and notifies the execution unit if a match between MPP and OPP is found. In this case, the execution unit then activates the corresponding plans.

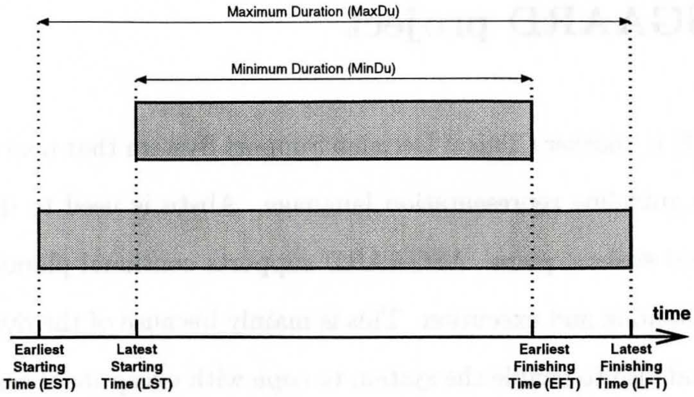
Maximum Duration (MaxDu)
Minimum Duration (MinDu)
time
Earliest Starting Time (EST)
Latest Starting Time (LST)
Earliest Finishing Time (EFT)
Latest Finishing Time (LFT)

Figure 2.2: The time scope of a plan in Asbru.

2.4.1 Asbru

Asbru plans have two parts, a data abstraction part that defines a set of parameters that are used in the guideline and a procedural hierarchy of plans. The plans hierarchy contains a set of plans that have a name, a set of arguments, a time annotation , preferences, intentions, conditions, effects, and plan body. The arguments are parameters that each plan receives from its parent plan (the one that invoked the plan). The temporal scope of the plan is represented using the time annotations. Figure 2.2 illustrates the time scope of a plan. The preferences describe the cost, resource constraints and the responsible actors. Intentions are the goals that the plan wants to achieve, e.g., keep the blood pressure below a certain threshold. Conditions represent the conditions that should be met between transitions of the plan state, e.g., to the active or complete state. The effects denote the relationships between measurable parameters and the plan input arguments by means of mathematical functions. The plan body finally contains a series of child plans to be executed in a specified order, i.e., parallel, sequential, ordered, or unordered.

ASGAARD represents the clinical knowledge as plans and does not support using data mining extracted knowledge. This is in contrast with our approach to use mined knowledge in the context of a flow chart based clinical guideline modeling language.

2.5 CHICA

Child Health Improvement through Computer Automation (CHICA) [15] is another Clinical Decision Support System that is developed to improve preventive pediatric primary care. It uses a knowledge-base of 290 *if-then* rules. The rules are encoded as separate Medical Logic Modules (MLM) using the Arden Syntax language. The main purpose of the system is to provide reminders to the care givers in order to enhance pediatric preventive care.

Two paper forms are dynamically generated by this system and are tailored to an individual patient. The forms are scanned and interpreted by the CHICA system in real time. One form contains questions that are answered by the patients prior to the patient-physician encounter, and acquires information about the patient, particularly the risk factors; and the second form delivers "just-in-time" reminders. Arden Syntax MLMs, known as "rules" in the system, are used to generate the content of these dynamic forms. In contrast to our approach, MLMs are not able to use mined knowledge.

2.6 PRESGUID

PREScription and GUIDelines (PRESGUID) [40] is a decision support system that integrates clinical practice guidelines with a drug data base and supports prescribing in primary care settings. Clinical guidelines defined as decision trees are coded in XML format and the system provides recommendations through a web based interface. In contrast to PREScription, our approach provides the knowledge in the form of data mining models, supporting decision tree models as well as a variety of other types of data mining models.

2.7 *e*-MS

e-MS [24] is not a Clinical Decision Support System. Instead, it addresses another problem in healthcare. An inherent characteristic of patient data is that it is usually scattered among many healthcare providers' offices and institutions. This necessitates proper communication and exchange mechanisms to be developed and deployed so that patient data sharing becomes a reality. Electronic Medical Summary (*e*-MS) is a project that intends to make subsets of patient data stored in one healthcare institution available to other stakeholders who want to have access to them. The *e*-MS project defines an XML-based Electronic Medical Summary (e-MS) document format, and an Electronic-Medical Summary Exchange Protocol (e-MSEP).

The e-MS documents are based on the HL-7 Clinical Document Architecture (CDA) release 2 standard [30]. The documents encode different pieces of patient data into a structured textual format. The document structure and semantics of different data items in the structure are available for users, they can interpret the meaning of the whole document. To be sent over the network, the clinical content should be wrapped by e-MS message wrappers that define another XML structure for the message. The whole XML structure is wrapped into e-MS SOAP (Simple Object Access Protocol) messages and sent over the network via HTTP. This structure contains of message content and the message header. The CDA document and some relevant MIME attachments reside in the message content.

e-MSEP on the other hand, is used for communication between healthcare entities and enables sharing of clinical documents, through a messaging framework and a message broker that supports message requests, responses, and queries. The data can be sent from one entity to other entities and the sender can track proper delivery of his message. The data items that are sent are delivered at the broker's drop point (message box) in the e-MS compliant format. The broker periodically polls the message box to get the received messages. Broker services are provided using the Web Services WSDL service definition language which also supports polling of the broker.

Table 2.3 summarizes the approaches of the systems discussed in this chapter.

Project name	Data access mechanism	Clinical best practice guideline modeling language	System architecture
EGADSS	CDA	Arden Syntax	Client-server
HELEN	Proprietary messaging format	HELEN guideline definition language	Client-server
COMPETE	Proprietary XML-based data exchange	Function tables	Client-server
ASGAARD	Proprietary XML-based data exchange	Asbru	local
e-MS	HL-7 Messaging and CDA	–	Client/Message Brokers via Web Services
CHICA	HL-7 Messaging	Arden Syntax	unknown
PRESGUID	–	Decision Trees	Web based client-server

Table 2.3: Clinical tools and projects

Chapter 3

Framework for interoperability of data and knowledge

The final goal of this research is to enable clinical decision support systems to gain access and the necessary power to interpret the data mining models that represent valuable medical knowledge. This type of knowledge is extracted in a data mining analysis. The resulting data mining models usually have a complex nature and the process of application of the models on new data requires careful handling of input parameters, as well as the output results. There are many steps involved in this process, and we are particularly interested in partitioning this process based on different logical tasks that involve both data and knowledge. Moreover, we precisely describe each partition and the data formats that flow in or out of each step.

After a short introduction, the rest of this chapter describes a framework for dissemination and application of mined knowledge; then we provide the details of the different roles and tasks that are involved in the framework by clearly describing the artifacts of each task, its inputs, and the corresponding stakeholders.

3.1 Introduction

In order to produce, disseminate, access, interpret, and finally apply mined clinical knowledge in a real world scenario different tasks are expected to be done by different stakeholders at different locations. Those who are involved range from healthcare researchers, healthcare providers, healthcare information managers, to healthcare knowledge managers, and clinical guideline modelers. The tasks are carried out by different healthcare institutions, universities, hospitals, or clinics using heterogeneous systems (e.g., Electronic Medical Record systems).

For the data and knowledge to flow smoothly throughout this complex process well-defined data and knowledge representation formats should be used. Since the interoperability of both data and knowledge are necessary, all producers and consumers of data and knowledge must comply with the commonly adopted standards. The framework provides an overall picture of the whole process. The details of achieving data and knowledge interoperability are explained in the subsequent chapters.

3.2 Framework for interoperability of data and knowledge

In this section, a framework for interoperability of data and knowledge is proposed that describes the process of extracting, encoding, and finally interpreting the knowledge that has been generated from mining healthcare data. Figure 3.1 illustrates the overall view of the proposed framework. The shaded area illustrates parts of the process that we have contributed to. Different tasks are carried out in the distributed healthcare environment by different parties with different expertise, knowledge, and skills. The framework consists of three phases:

- Preparation: the healthcare data is mined and useful patterns and trends are ex-

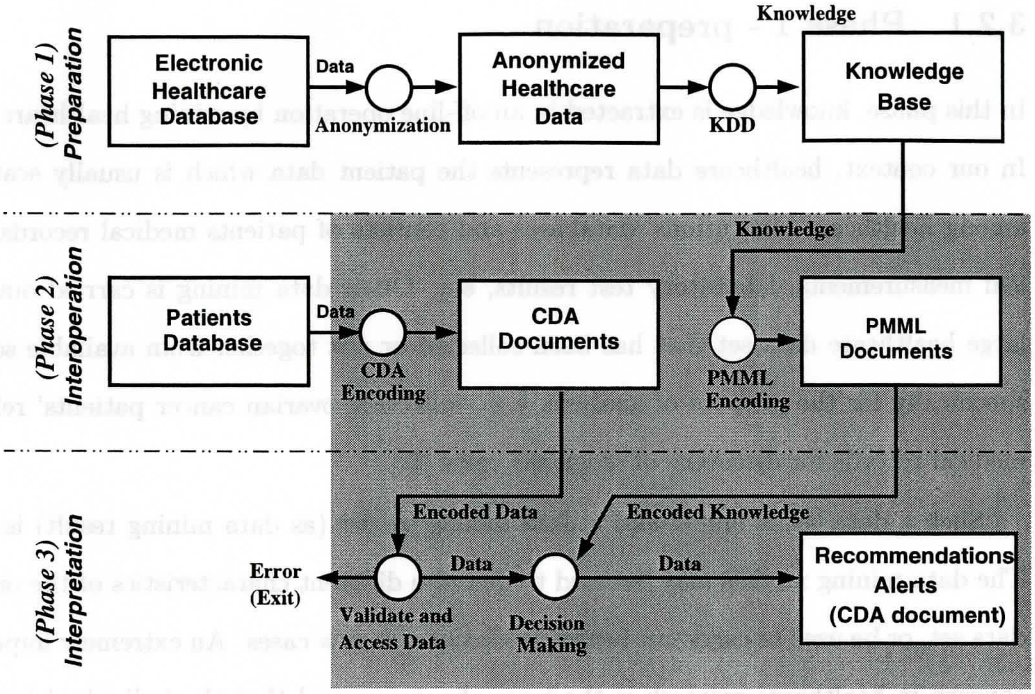

Figure 3.1: Healthcare framework for interoperability of data and knowledge. The shaded area designates the activities that are based on our approach in data and knowledge interoperability.

tracted from the data set in the form of data mining models. In this thesis we assume that this phase is done by expert medical researchers.

- Interoperation: the mined knowledge is encoded in an XML-based format that is understandable by different parties. The patient data is also encoded using the healthcare data interoperability standards.

- Interpretation: the mined knowledge is applied to describe characteristics of patient data that were discovered, or to perform predictions for the new data cases based on the extracted results.

A more in-depth description of each phase follows.

3.2.1 Phase 1 - preparation

In this phase, knowledge is extracted in an off-line operation by mining healthcare data. In our context, healthcare data represents the patient data which is usually scattered among healthcare institutions' databases and consists of patients medical records, clinical measurements, laboratory test results, etc. Often data mining is carried out on a large healthcare data set that has been collected or put together from available sources specifically for the purpose of analysis, e.g., collecting ovarian cancer patients' relevant medical records for discovery of diagnosis rules [42].

Such a data set is mined and a data mining model (as data mining result) is built. The data mining models may be used to describe different characteristics of the original data set, or be used to carry out future predictions on new cases. An extremely important concern in healthcare research is the issue of privacy, and that the individuals' privacy should be protected by all means [45]. The healthcare data collection should *only* contain the minimal set of data items that are required for the purpose of analysis. Patients real identifiers, e.g., names, Social Security Numbers, are usually considered as *not* required fields, and as a common privacy protection practice, they are excluded from the collected data set.

The next step, Knowledge Discovery in Databases (KDD) is carried out on the anonymized data. KDD itself has several different steps, data selection, data preprocessing, data transformation, data mining, and evaluation of results. We will briefly review these steps in Chapter 4. The knowledge discovery activity is often time consuming and needs to be guided/assisted and the results need to be evaluated in many ways by expert medical researchers, statisticians, and scientists. At the end of the KDD process, the results that contain valuable (mined) knowledge are stored locally in the knowledge-base. Examples of such data mining models are used to classify patients for diagnosis based on different physical symptoms [10]; cluster the patients based on relevant risk factors [20]; and extract useful and hidden patterns in data as in the case of

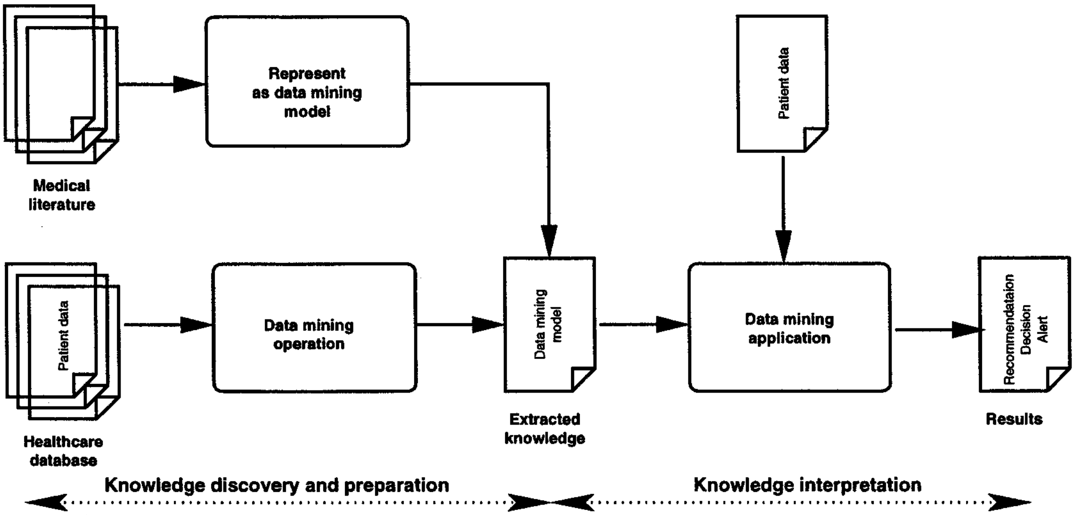

Figure 3.2: Different activities to build and use data mining models.

association rules mining [39].

We can further broaden the definition of knowledge to any type of knowledge that can be represented as a data mining model. In this case, the source of the knowledge can be the medical literature and need not be a data mining extraction process. By this definition, we add the support for a variety of clinical examples that can be represented by data mining models in our framework. We consider the activities that are carried out in this phase (the preparation phase) as the off-line data mining model building activity. This is in contrast with the online application activity during which the mining models are interpreted and applied on new patient data. Figure 3.2 illustrates these two actitivities.

3.2.2 Phase 2 - interoperation

In the second phase of the framework we are mainly concerned with the activities that are required to make the mined knowledge (from phase 1) available to interested users. These stakeholders may use different applications, databases, and information systems that are heterogeneous to the ones used by the institution that initially carried out the data analysis and knowledge extraction. As a result, this phase involves taking care of encoding both data (the data mining model's input or output) and mined knowledge. This phase ensures the interoperability among the software systems of different institutions.

Interoperability of data

To lay a foundation for data interoperability we adopt a standard healthcare data model to be shared between senders and receivers of patient data. A common data model provides a shared view of healthcare data to the heterogeneous healthcare software systems. Using this data model, patient's data items are mapped from internal data representations to the corresponding data fields in the data model. In our research, we use an XML-based standard called Clinical Document Architecture (CDA) [30] to encode a patient's data in order to allow data interoperability. The CDA standard specifies development of XML schemas that are shared among senders and receivers. Heterogeneous information systems can communicate properly by authoring structured XML documents according to the CDA schema. The specification is based on a common data model that is referred to as the Health Level-7's Reference Information Model (HL-7 RIM). The development of the CDA schema in our framework is done off-line, however, the CDA documents are generated online. The sender associates each data item with a particular semantic location in the CDA document's structure; and the receiver accesses the data item in the same way.

Similar to the patient data, we have to provide interoperability of the results of decision making too. These results may contain patient-specific recommendations, alerts, reminders, etc. For this purpose, we use the CDA documents to provide semantic data interoperability of the results. The resulting CDA documents are again generated online and according to the CDA schema and specification. Further details related to encoding and accessing patient's data items, along with validation of CDA documents are described in Chapter 5.

Interoperability of knowledge

At the end of phase 1 (3.2.1) the final results of data mining were stored using proprietary encodings (supported by the mining tools). Hence, as a first step for knowledge interoperability we re-encode the mining models using a standardized XML-based markup language, namely Predictive Model Markup Language (PMML) [21]. PMML provides the required constructs to precisely describe different elements, input parameters, model specific parameters, transformations, and results of a variety of types of data mining models. The details of encoding the mining models are deferred to a chapter on the interoperability of data mining results (5). Details of the data/knowledge related activities in this phase are elaborated collectively in a separate chapter on interoperability of data and data mining results (5).

3.2.3 Phase 3 - knowledge interpretation

The third phase, knowledge interpretation, uses the mined knowledge that was discovered in phase 1, and was prepared for dissemination and application in phase 2. The knowledge is interpreted for specific patient data that is encoded in a CDA document. The interpretation is done by a *logic module* which is a program that is capable of parsing both the CDA and PMML files to get the case data and to eventually apply the mining model on the data items. Based on the results of this application a final decision is made.Figure 3.3 illustrates the different steps that are performed in this phase. A short description of these steps follow:

1. Retrieving the CDA document instance from the EMR system: the CDA schema that was developed in phase 2 is used as a template for retrieving the case data from the EMR system.

2. Validate the CDA document: the CDA document instance is validated against the CDA schema and additional constraints that are defined. If no violations are found

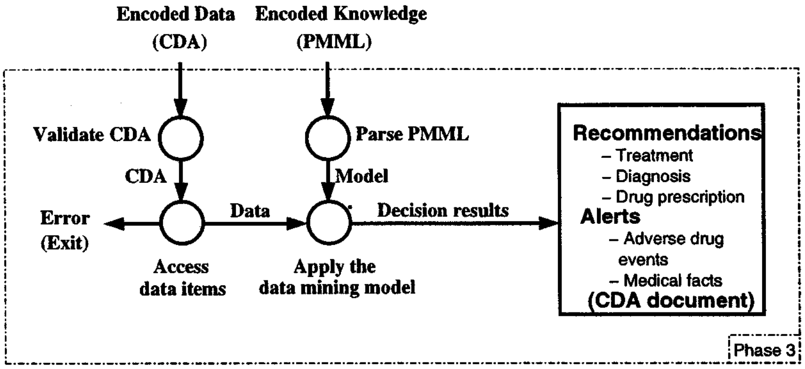

Figure 3.3: Different steps involved in the third phase of the framework (knowledge preparation).

the process continues to the next step.

3. Access the data items that correspond to the inputs of the data mining model: in this step, the CDA parser accesses and retrieves the data items from the corresponding locations in the CDA document structure.

4. Interpret the PMML model: the data mining model is applied to the patient data. Each logic module is capable of parsing the corresponding PMML document, constructing the mining results data structures for a particular type of mining models, and applying the model.

5. Encoding of the results: encode the results in CDA results documents.

It is important to note that phase 3 is done online at the usage site (i.e., point of care) by users, rather than the party who actually performed the mining analysis. Further details of how these steps are performed in our framework are described in Chapter 5, interoperability of data and knowledge.

3.3 Different stakeholders involved in the framework

The activities in the framework are usually carried out in a decentralized and collaborative healthcare environment in which different parties (i.e. healthcare organizations, researchers) are involved. Hence a clear tasks separation and specification is required to be defined. The tasks fall into 5 groups:

- **Data collection**: data collection is carried out by the healthcare researchers. There is a pre-specified and fully defined purpose for the process of data collection that specifies which data items should be subject to collection. Regulatory and privacy laws in effect in the country, province, or region may apply. It is usually the case that related governmental health agencies and authorities, and the ethics board of the health organization that carries out the data collection should ratify the purpose of collection.

- **Data mining analysis**: the analysis of data extracts useful and novel patterns from the data and builds models that describe the data or are used for predictions on future case data. The data mining algorithms and mining tools are usually developed and fine-tuned for the purpose of analysis by computer scientists. The data analysts and statisticians are expert in guiding the knowledge discovery process through possibly many iterations and evaluation of the results. They also select strategies on how to handle invalid or missing data items.

- **Data interoperability related activities**: each stakeholder may have different internal data representations. Healthcare data modelers define a common data structure that is used as a common model for data exchange. IT staff provide institution specific mapping to the common data models.

- **Knowledge interoperability related activities**: the mined knowledge has to be made interpretable and accessible to different users other than the party who carried out the data mining analysis. Computer scientists build standard and extensible data mining models that can be used to encode data mining results. Along with IT developers and programmers they build logic modules that are capable of interpreting the mining models to yield a decision. Data analysts also provide valuable information on the internals of each particular mining model that is required for the purpose of encoding.

- **Application**: the usage happens at the point of care in the healthcare providing institutions. The patient is subject to the decision making and is the source of (owns) the case data. The healthcare professionals (i.e. medical practitioners) use the CDSS to access the mined knowledge for assisted decision making.

Table 3.1 summarizes different tasks in the framework and specifies the location and the parties that are responsible to carry out the related activities.

Performer	Tasks	Location
Healthcare professionals, medical practitioners	Accessing the logic modules for assisted clinical decision making through the decision support system	Point of care healthcare delivery
Computer scientists, data analysts, programmers, IT developers	knowledge interoperability related tasks	Healthcare research centers
Data modelers and IT staff	Data interoperability related tasks	Healthcare institutions
Data analysts, computer scientists, statisticians, and expert medical researchers	Perform the data mining studies	Healthcare research centers, universities and medical schools
Governmental and local healthcare authorities	Ratification of the purpose of data collection	Government agencies and local ethic boards of the health institutions
Healthcare personnel	Data collection	Health institutions

Table 3.1: Different stakeholders involved in the framework and their corresponding tasks.

Chapter 4

Knowledge extraction

The main purpose of this research is to make the mined clinical knowledge available for use by interested users. In this context, data mining operations are used to extract novel, useful, and non-trivial knowledge from healthcare repository. In our framework, this process takes place in the first phase (preparation). Figure 4.1 illustrates different steps that are carried out in the preparation phase.

In this chapter, we first give an overview of some concerns regarding healthcare data privacy issues. Then we review the data mining driven knowledge extraction process and describe some of the data mining techniques that are commonly used. We will then give application examples of data mining analysis on healthcare data from the literature. Finally, the chapter ends with description of a case study which will be used as a running case study in the following chapters.

4.1 Healthcare data privacy

An extremely important concern in healthcare research is to protect the patient's data privacy by all means [45]. Privacy protection laws are currently in effect for this purpose in many countries including Canada, United States, Japan, and Europe. In essence all of these countries impose strict measures and regulations governing the collection, and

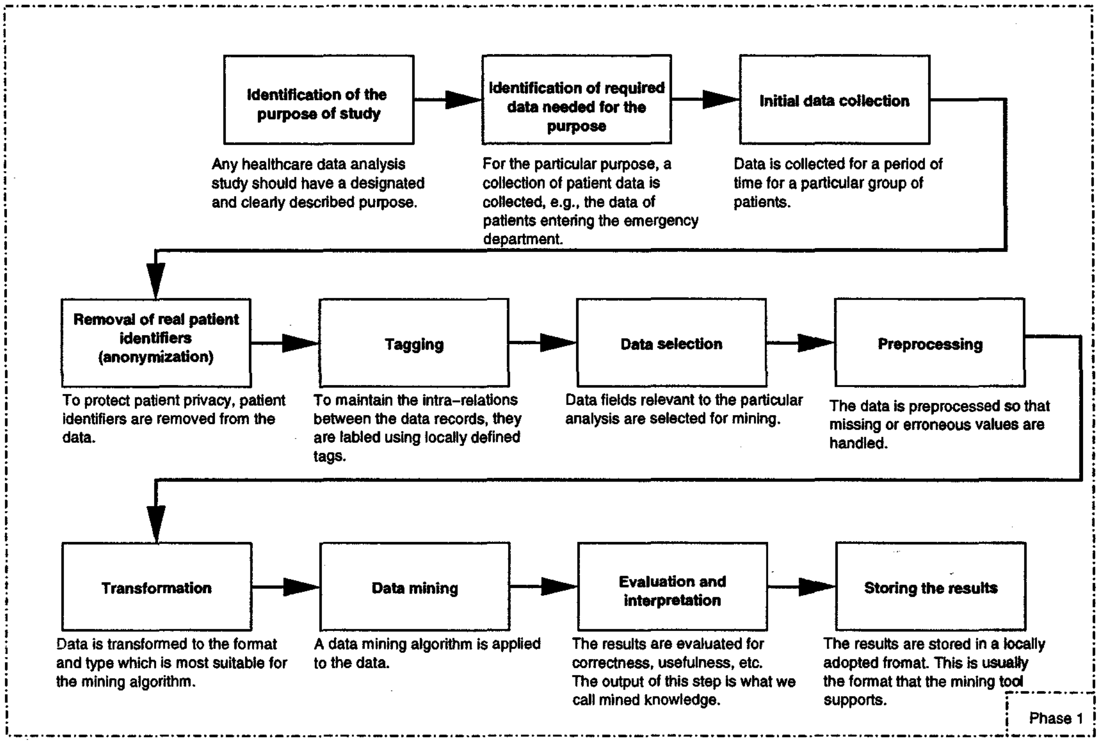
Identification of the purpose of study
Any healthcare data analysis study should have a designated and clearly described purpose.
Identification of required data needed for the purpose
For the particular purpose, a collection of patient data is collected, e.g., the data of patients entering the emergency department.
Initial data collection
Data is collected for a period of time for a particular group of patients.
Removal of real patient identifiers (anonymization)
To protect patient privacy, patient identifiers are removed from the data.
Tagging
To maintain the intra-relations between the data records, they are labled using locally defined tags.
Data selection
Data fields relevant to the particular analysis are selected for mining.
Preprocessing
The data is preprocessed so that missing or erroneous values are handled.
Transformation
Data is transformed to the format and type which is most suitable for the mining algorithm.
Data mining
A data mining algorithm is applied to the data.
Evaluation and interpretation
The results are evaluated for correctness, usefulness, etc. The output of this step is what we call mined knowledge.
Storing the results
The results are stored in a locally adopted fromat. This is usually the format that the mining tool supports.
Phase 1

Figure 4.1: Different steps involved in the first phase of the framework (knowledge preparation).

use of patients' private data. A common theme in these privacy laws is that any patient data that is collected should bear a legitimate purpose which also defines and confines the usages of those records. The person who is subject to the data collection should in turn be fully informed of the subsequent usages and applications in future analyses of his/her data. Patients' consent is hence crucial for the data to be collected, stored, and analyzed. The collection should *only* contain the minimal set of data items that are identified to be required for the purpose of analysis. Also, before any data collection can take place, the governmental and/or local healthcare authorities and the ethics board of the collecting institutions should ratify the purpose of the data collection according to the specific laws that apply.

Patients' real identifiers, e.g., names, Social Security Numbers, are usually considered as unnecessary, and as a common privacy protection practice, they are excluded from the collected data set. However, in many cases a single individual may have several different records in the data set and removal of the patient identifiers results in loss of these interrelations between data records. To enable tracing such interrelated records, a fast solution is to tag records with local identifiers prior to anonymization. The anonymized data set is the input to the knowledge discovery process.

4.2 Introduction to knowledge discovery in databases

Knowledge Discovery in Databases (KDD) is the process to extract hidden relationships in large databases. The KDD process is interactive and iterative, and involves several steps: data selection, data preprocessing, data transformation, data mining, and evaluation of results [27].

1. **Data selection:** first, the data fields of interest are selected. The selection is in fact a proper subset of the data attributes that were collected in the data collection activity. For instance, in data collection the research team might choose to monitor

or collect data from all of the patient's physical examinations, while in this step, particular fields, e.g., weight or height measurements are selected to be used for the data mining operation.

2. **Data preprocessing:** it is often required to do certain preprocessing on the data, and take certain measures to handle erroneous or missing data items. The preprocessing step involves checking data records for erroneous values, e.g., invalid values for categorical data items, and out of range values for numerical attributes. In the real world practice, many records may have missing values. The researchers may decide to exclude these records from the data set, or substitute missing attributes with default or calculated values, e.g., the average of the values in other records for a missing numerical attribute.

3. **Data transformations:** The data items in their raw format are normally not suitable for mining. Several different types of transformations may be applied to the data items to make them more appropriate for the particular purpose of mining. The transformations can be considered as changing the basis of the space in which data records reside as points in this space. For example the patient's weight in millimeter has probably too much precision; hence a conversion to centimeter or meter may be considered. Additionally, the data mining expert may choose to transform the weight value into discretized bins to further simplify things for the mining process. Also, there might be some fields that are derived from other data attributes, e.g., the duration of an infection can be derived by subtracting the initial diagnosis date from the date that the treatment was completed.

4. **Data mining:** In this step, a data mining algorithm is applied to the data. The choice of the algorithm is decided by the researchers and depends on the particular type of analysis that is being carried out. There are a wide range of algorithms available, but we can group them into two categories: those that describe the data,

or those that do predictions on future cases [13]. As briefly described in the Section 4.3, the algorithms can also be grouped based on the type of mining they perform, e.g., clustering, classification, and association rules mining are examples of these types.

5. **Evaluation and interpretation of results:** it is essential that the results be evaluated in terms of meaningfulness, correctness, and usefulness. Based on the evaluation of results, the researchers may choose to go some steps back and perform them again differently. This makes the knowledge discovery process an iterative process. After completion of the discovery process, we refer to the extracted results as mined knowledge. These results are eventually stored in some application (data miner tool) specific format for future access and use.

4.3 Data mining models

Data mining models are data structures that represent the results of data mining analysis. There are many types of data mining models. In this section we briefly describe some major types, classification, clustering, and association-rules models. There are numerous algorithms in each category that typically differ in terms of their data or application specific fine tunings, their performance and approach in building the models, or the case- or domain-specific heuristics they apply to increase the efficiency and performance of the mining process.

As far as we are concerned in our framework, we don't differentiate between different implementations and algorithms of any of the data mining categories, if their results can be represented by the general constructs of the corresponding data mining type. For instance, different association rules discovery algorithms take different approaches in extracting the frequent item sets and opt to choose different measures to exclude intermediary sets and hence prevent explosion in the results set. Some may refine the set based on standard constraints of support and confidence, others may apply additional constraints on the size of the rules' antecedent and consequent.

4.3.1 Classification models

A classification algorithm (e.g., neural network or decision tree) assigns a class to a group of data records having specific attributes and attribute-values. The classification techniques in healthcare can be applied for diagnostic purposes. Suppose that certain symptoms or laboratory measurements are known to have a relation with a specific disease.

A classification model is built that receives a set of relevant attribute-values, such as clinical observations or measurements, and outputs the class to which the data record belongs. As an example, the classes can identify "whether a patient has been diagnosed with a particular cancer or not", and the classifier model assigns each patient's case to one of these classes.

4.3.2 Association rules models

Association rule $X \Rightarrow Y$ is defined over a set of transactions T where X and Y are sets of items. In a healthcare setting, the set T can be the patients' clinical records and items can be symptoms, measurements, observations, or diagnosis. Given S as a set of items, $support(S)$ is defined as the number of transactions in T that contain all members of the set S. The *confidence* of a rule is defined as $support(X \cup Y)/support(X)$ and the support of the rule itself, is $support(X \cup Y)$.

The discovered association rules can show hidden patterns in the mined data set. For example, the rule:

$$\{People\ with\ a\ smoking\ habit\}$$

$$\Rightarrow \{People\ having\ heart\ disease\}$$

with a high confidence; might signify a cause-effect relationship between smoking and the diagnosis of heart disease. Although, this specific rule is a known fact that is expected to be valid, there are potentially many more rules that are not known or documented.

4.3.3 Clustering models

The last group of data mining techniques that we describe in this section is clustering. Clustering is originated from mathematics, statistics, and numerical analysis [18]. In this technique the data set is divided into groups of similar objects [18]. The algorithms usually try to group elements in clusters in a way to minimize the overall distance measure (e.g., the Cartesian distance) among the cluster's elements. Data items are then assigned to the clusters based on a specific similarity measure. And researchers then study the other properties of the generated clusters.

4.4 Data mining applications in healthcare

Healthcare data mining analysis produces valuable knowledge that can be used for decision making. Various types of mining models (e.g., clustering, classification, and association rules models) can represent different types of hidden patterns and trends in clinical data with numerous applications in medical practice. In this section we briefly review some of the existing applications in the literature.

Churilov et al. [20] describe a clustering method using an optimization approach to extract risk grouping rules for prostate cancer patients. The data record fields are the patients age, tumor stage, Gleason score, and PSA level (in this paper the medical meaning of these fields are not of our interest). The clustering algorithm generates 10 clusters that are then grouped to low, intermediate and high risk categories. Ordonez et al. [39] propose a new algorithm to mine association rules in medical data with additional constraints on the extracted rules and applies the method for predicting heart

Association Rule	Support	Confidence
$SeptoAnterior \Rightarrow (LAD \geq 50\%)$	18%	80%
$InferoSeptal \Rightarrow (RCA \geq 50\%)$	12%	65%
$InferoLateral \Rightarrow (LCX \geq 50\%)$	20%	53%

Table 4.1: Most significant discovered association rules in mining heart disease data [39].

disease. Evaluation shows that most significant rules are also verified by expert medical practitioners, and three of the most important ones are represented in Table 4.4.

A decision tree-based classification approach has been applied to mass spectral data to help diagnosis of ovarian cancer suspects [42]. While association rule classifiers have been applied to diagnose breast cancer using digital mammograms [47]; Land et al. use a Neural Network based classification approach for the same purpose [34]. Li et al. [36] discuss the problem of mining risk patterns in medical data using statistical metrics in the context of an optimal rule discovery problem and apply the method to find patterns associated with an allergic event for ACE inhibitors. Association rules mining is also applied over data of human sleep time [35]. Wilson et al. [46] discuss potential uses of data mining techniques in pharmacovigilance to detect adverse drug reactions. Duch et al. [10] compare various data mining methods supporting diagnosis of Melanoma skin cancer. The last study mentioned above serves as the case study for this research that is discussed throughout the rest of the thesis.

4.5 Running case study

In this section, we describe a healthcare data mining analysis research from the literature that we have used as our case study in this thesis. This case study refers to a classification data mining analysis that has been carried out by Duch et al. [10] on patients' data. The classifier is a decision tree that classifies patients into four types of Melanoma: benign,

Data item	Accepted values
Asymmetry	Symmetric-spot = 0, 1-axial asymmetry = 1, 2-axial asymmetry = 2
Border	Values from 0 to 8
Color (Binary coded)	White, Blue, Black, Red, Light brown, Dark brown
Diversity (Binary coded)	Pigment globules, Pigment dots, Branched strikes, Structureless areas, Pigment network
$C - Blue$	Absent = 0, Present = 1

Table 4.2: Description of the different data items accessed by the decision tree classifier.

blue, suspicious, or malignant. The data have been collected in the Outpatient Center of Dermatology in Rzeszw, Poland containing 250 records.

The data selection for data mining contains five variables, indicating presence or absence of $C - Blue, asymmetry, border, color$, and $diversity$ of the skin cancer mark's structure. The latter four variables are used to calculate an index, called Total Dermatoscopy Score (TDS). The TDS index is calculated by the following formula:

$$TDS = 1.3 * Asymmetry + 0.1 * Border + 0.5 * \Sigma Colors + 0.5 * \Sigma Diversities \quad (4.1)$$

Table 4.2 describes the different data items that are used in calculating TDS.

The data mining operation has been carried out on the calculated TDS and C-Blue variables to build the decision tree classifier. Figure 4.2 illustrates the resulting data mining model.

We will use this example data mining model in the following chapters as a running case study to explain different activities that are done in our framework to make the classifier available in Clinical Decision Support Systems.

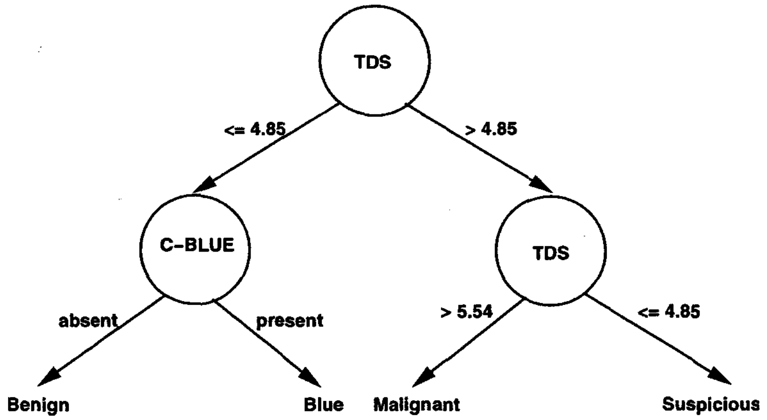

Figure 4.2: The decision tree classifier for Melanoma skin cancer.

Chapter 5

Interoperability of data and mined knowledge

Currently, Information Systems (IS) have been deployed by many healthcare organizations for a wide range of different purposes, including but not limited to telemedicine, patient care, Electronic Health Record (EHR) systems, clinical or administrative decision support and many more. Many scenarios require the patient data that is stored in one such system to be made available for other healthcare institutions who are interested to access it. A major obstacle to the widespread use of IT in healthcare settings is the high degree of heterogeneity between healthcare Information Systems. Since different institutions often use different formats to store identical pieces of data internally, it is challenging for these communicating systems to have the same understanding of the same data.

The patient data usually crosses the systems boundaries in the form of well formatted messages. Also, a messaging framework is usually developed and deployed to handle the flow of data messages. The framework may support reliable message delivery by having acknowledgements sent back to the sender to inform correct or erroneous receipt of the message. It may provide security or information confidentiality support, encryption services, or events handling infrastructure (e.g., publish/subscribe model [33]) where a receiver subscribes to receive a message when an event of interest occurs. The implementation technology (e.g., Java Remote Method Invocation (RMI), Remote Procedure Call (RPC), Web services using SOAP) and the format of the messages (e.g. MIME, XML-based, or proprietary ASCII formats) may differ in each framework, or a single framework may support different types of message format. The messaging infrastructure itself relies on the transport protocols (e.g., FTP or HTTP), and the underlying communication channel (e.g., internet) between information systems.

Also, central in our framework (chapter 3) is the process of making data mining results (data mining models) accessible and interpretable by different parties. We refer to this ability as knowledge interoperability. Knowledge interoperability relies on data interoperability to materialize seamless dissemination and application of mined knowledge. In the second part of this chapter we will elaborate on our approach to achieve this goal. We adopt an XML-based encoding standard which encodes the data mining models as XML documents. The documents can be shared, exchanged, and interpreted by healthcare systems at different institutions. Specifically data items are retrieved from the Electronic Medical Records System or from the user input, and knowledge is accessed through XML documents that are eventually interpreted for clinical decision making. The results of this application can then be used for making a decision or assisting the healthcare professionals by providing them with alerts or recommendations through the CDSS user interface.

5.1 Introduction to data interoperability problem in healthcare

The healthcare environment constitutes a network of information systems that are connected. In this network, each node represents a computer system that is deployed and maintained by a healthcare institution. For example, one node may be a clinician's PC in his clinic that stores the medical records for his patients. Another node may run the patients management software for managing receptions in an emergency department, or act as the central Electronic Medical Records (EMR) system of a hospital.

In many cases, the data in one such system has to be accessed by other nodes. Example scenarios that require this kind of data access are when a patient referral to another healthcare institution takes place, or when the patient moves to other locations and wants to have their medical record data be moved to a nearby clinic as well. Patients' medical records contain different types of information including but not limited to the medications that he is receiving, his medical history, the results of lab tests, physical examinations, observations and measurements, and even relevant data items from his relatives medical data (e.g., the cause of death for his father was lung cancer).

The heterogeneity of information systems in the distributed healthcare environment adds more challenge to the task of data managers. Since the sending and receiving information systems are very likely to have different internal data models, it is not enough to simply send and receive messages and parse the messages to get the encoded bit streams. The receivers should understand the meaning associated with different data pieces in the message as well. Data interoperability refers to this desired property of heterogeneous systems to be integrated seamlessly and collaborate by making effective use of information. The data interoperability can be decomposed in two parts, syntactic and semantic interoperablity.

5.1.1 Syntactic data interoperability

Syntactic data interoperability is the ability of information systems to communicate using the same terms to refer to identical concepts. For an example of lack of syntactic data interoperability, consider one system that refers to 'coughing' as *cough*, and another system that refers to the 'coughing' as *Husten* (which is the German translation of cough).

Further consider the first system retrieves the clinical record of a particular patient from the second system. If there has been any reference to 'coughing' in the received patient files, the first system cannot understand it since it refers to 'coughing' with a different name. Hence, any reference to *Husten* in the messages will be discarded.

Syntactic data interoperability is achieved by adoption of common vocabulary sets. These vocabularies can be developed and used locally, or an external one can be adopted. LOINC (Logical Observation Identifiers Names and Codes) [5], UMLS (Unified Medical Language System) [11], SNOMED CT (Systematized Nomenclature of Medicine Clinical Terminology) [8], ICD (International Classification of Diseases) [1], and MeSH (Medical Subject Headings) [7] are among the most important medical terminology systems that are currently available. Each vocabulary set is also identified with an identifier. Similarly within each vocabulary system, each term is uniquely identified by its identifying code.

5.1.2 Semantic data interoperability

In addition to the syntactic data interoperability, semantic data interoperability is another building block for enabling seamless data exchange between heterogeneous information systems. Semantic refers to the meaning that the information systems perceive from each term. Semantic data interoperability is the joint ability of the sender and receiver of data to convey the full semantic and context in which each term receives a particular meaning. In the healthcare domain, there are a vast collection of concepts [1] that in many cases have shared names (referred to with identical terms). For instance consider the coughing example from the previous section and suppose that both systems now use the term *cough* (syntactic data interoperability in place) to refer to the general concept of coughing. If the first system receives this term in a message, it should also be able to understand what is meant by that, as this term by itself does not convey any useful meanings. *Cough* can be a symptom of a disease with a description in the

[1] Millions of concepts.

medical literature; an observation in the patient's clinical history data; or even the cause of death of his parent! It is important that this associated context be conveyed to the receiver in a proper way so that it can recognize the meaning of the term *cough* for correct interpretation of the received messages.

5.1.3 Common data model approach

One way to achieve the desired data interoperability is to adopt a common data model to model, represent, and encode health related data. This data model provides a unified view among the senders and receivers of data to communicate effectively and interpret the encoded data with the same meaning at both ends of the communication channel.

For instance, consider the coughing example once again. The sender already knows the context and the semantics associated with the term *Cough* as stored in its databases using its own internal data representation models, e.g. a record in a relational database table. This internal data structure is likely to be different from the data models at the receiver site that represents the same data semantic using a different format, e.g. a different relational schema. Before packaging the data in a message, the sender maps its internal data to the common data model, encodes and packages the message, and finally sends it off on the network to the receiver. At the receiver, it will receive and decode the message. The content of the message will then be mapped from the common data model to its internal data models. If the mappings are performed correctly the meaning is preserved.

5.1.4 Pair-wise data mapping approach

The common data model approach contrasts with the pair-wise data mapping approach in which the sent data is mapped directly to the receivers' internal data representation formats. This approach has a complexity of $O(N^2)$ for a network of N participating heterogeneous information systems. It is also ambiguous who (the sender, the receiver,

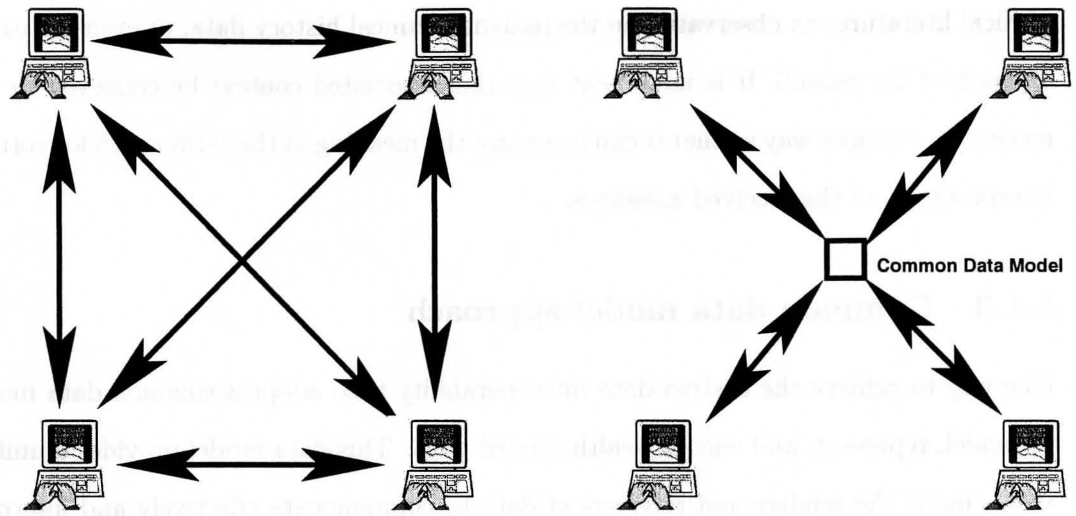

Figure 5.1: Pair-wised mapping and centralized mapping approaches for data interoperability.

or a third party) is responsible to perform the data mappings. In any case, the parties that perform the mapping need to have extensive knowledge of how data is represented internally at each institution. This makes the approach very hard to implement and in many cases infeasible. On the other hand, the common data model approach has a complexity of $O(N)$ which requires much less effort to be put in performing the necessary mappings. It is generally the responsibility of the sender and the receiver of the messages to provide the required mappings to the common data representation, and back to their internal data models respectively.

However, the common data model should be developed with generality in mind to be able to encode different and numerous possible semantics. This makes both the development of the model and the mapping of data items a much harder task, since the data modelers and data managers are dealing with a wide range of classes as opposed to limited, locally adopted models. Figure 5.1 illustrates the two approaches.

5.2 Standard-based data interoperability

Healthcare researchers have been active in the development of a common data model for the healthcare domain. Health Level-7 (HL-7) [2] is a Standards Developing Organization consisting of an international community of healthcare experts and information scientists. It has been active in this arena by developing and promoting the use of standards for the exchange, management and integration of electronic healthcare information. Many of its standards have been approved by the American National Standards Institute (ANSI).

To achieve the required data interoperability in our framework, we have adopted the HL-7 Clinical Document Architecture (CDA) standard [30]. The standard is based on the HL-7 Reference Information Model (RIM) [31] and is used to define structured XML documents that can encode patients clinical data. CDA is a broad specification and can be used to represent complex relations between data elements. Data input to the mining models are encoded in relevant CDA documents and the CDSS will retrieve and parse the documents to eventually access the data items. After doing the job of applying the mined knowledge and making a decision, the CDSS will output the results in the form of recommendations or alerts. The results will also be encoded in the form of CDA documents that refer back to the input documents. The results are also interoperable and may be stored at a patient medical records system for future access and processing, or be displayed to the user.

The data mining researchers who built the data mining models, specify the requirements of the different input data to each mining model as well as the output values. These specifications are then used to develop the CDA document schemas, subsequent CDA document instances, and a validation document containing a set of constraints over the data. We represent the data mining model specific input constraints in separate XML documents. The validation documents along with the CDA schema documents and the mining model's data requirements specification form the different bits and pieces that enable data interoperability in our framework. These documents should be ported to the usage site so that the received CDA documents (containing patient data) can be validated and parsed correctly.

5.2.1 CDA schema document

Having the data mining model's input/output data specification, we have to specify how to encode the data within the CDA structure. At this point we generate or adopt one or more XML schema documents that define the CDA structure that can contain the data mining input/output data items. Each location in the schema is associated with a particular semantic that is defined by the CDA specification and the underlying HL-7 RIM.

The CDA specification distinguishes three levels of details to be encoded in the final documents.

- **Level 1** — The first level, declares a header for the conforming documents that encodes the general description of the document, its purpose, the information about the participating entities (e.g., healthcare institutions, hospitals, clinics, doctors), the patient general identification information.

- **Level 2** — The second level is built on top of what the first level provides and extends it to include clinical data in the form of structured text blobs. The structured blobs are scattered islands in the XML document and can be rendered and displayed to the human users. At this level, CDA provides constructs and XML elements that can be used to format the information in a human readable way.

- **Level 3** — The third level provides XML elements and attributes to encode details of the patient clinical data in a fully structured and computer interpretable manner. In many cases, a structured text blob will accompany a clinical data segment in level 3 that represents the same information in a human-friendly and renderable manner.

Since we are interested in having the CDA documents be used directly by computer programs (i.e. a CDSS) we will be using the CDA specification at level 3 which also includes both levels 1 and 2, and we elaborate on how to access data items as they are scattered in the CDA.

Development of the CDA schema is a difficult and challenging task. This is mainly because the healthcare domain model (RIM) has a large number of classes, relationships, data types, and coded values. For this reason CDA schemas are usually developed with generality in mind so that they can be used for different purposes. This is in large realized by using coded values to define the scope and meaning of many XML elements that have a general meaning. For example, the semantic associated with the descendant elements under `<observation>` can be specified by its `moodCode` attribute. Possible values for the `moodCode` specify whether the observation is an event that has happened (`EVN`), definition of an observation (`DEF`), goal and objective (`GOL`), or if the observation is intended or planned (`INT`), a commitment or promise (`PRMS`), a proposal to perform the entry (`PRP`), or a request to perform the entry (`RQO`).

The attribute `moodCode` is present in many elements within the CDA documents, e.g., `<procedure>` (to encode a procedure or surgery or treatment), `<substanceAdministration>` (to encode administration of different medications), `<observationMedia>` (to encode references to medical images), and `<encounter>` (which represents a patient encounter with clinical institutions or personnel, e.g., hospitalization, referral). The `<observation>` element itself encodes a variety of concepts from lab results, physical examinations, failure or success of treatments, allergies, alerts, risks, etc. Having a large set of semantic elements at hand, we note that the exact details of how to encode a particular data item has to be addressed in a case by case manner.

In our case study (presented in the appendix), we have used a CDA schema that has been developed in the context of the *e*-MS project. This schema is able to represent patient identification information, lab results, patient medical history data, patient social history/risks, physical examination measurements, patient's active problem list, current medications, surgery and medical imaging history, treatments, immunizations, allergies, and family history data [25].

While the CDA schema (along with other incorporated schemas, i.e., data types schema, vocabulary schema) lay the foundation for semantic data interoperability, it does not address the syntactic data interoperability directly. The schema only provides basic attributes to specify each term's `code`, `codeSystem`, `displayName` and `codeSystemName` to associate values in the CDA document with external vocabulary sets that have to be developed, distributed, and used separately. The location of a piece of data within the structure along with the values of the XML elements and attributes represents the semantic that the source intended to convey.

5.2.2 CDA instance document

CDA document instances are XML documents that actually contain patients' clinical data. They conform to the associated CDA schema that was developed and distributed among the healthcare institutions participating in data exchange for different purposes, e.g., sharing the results of laboratory tests, or patient discharge summaries. In the proposed framework CDA documents both encode the input to the mining models as well as the output results after application of the mining models. The output document can reference the input CDA documents using their identifiers in the header section of the CDA. The input CDA instance will be built by the data source owner and the output CDA instance is generated by the logic module of the CDSS. Since the results are data mining specific, we defer the details of encoding them to the corresponding section in this chapter.

5.2.3 Accessing data items in the documents

To access data items in the CDA instance documents, a parser should parse the XML structure and look for locations of interest that hold the relevant data elements. For this purpose we use the *XPath expressions.* XPath is a language for addressing different parts within an XML document. In our usage, we associate with each data item of the data mining model's input/output, a corresponding XPath from the input/results CDA document. For the input values, the XPath engine will eventually look up and select the node(s) or data value(s) that the XPath expression refers to. The output values will also be written to their associated XPath address in the output CDA document. This way, the semantic associated with the selected data values is encoded in the XPath expression.

The following XPath example selects the value of “*Troponin I*”[2] from the laboratory results for patient with last name “Sherafat”.

```
/hl7:ClinicalDocument/hl7:recordTarget[@typeCode='RCT' and
@contextControlCode='OP']/hl7:patient[classCode='PAT']/
hl7:patientPatient/hl7:name[hl7:family='Sherafat']/
ancestor::hl7:ClinicalDocument/hl7:component/hl7:structuredBody/
hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
hl7:observation[@moodCode='EVN' and classCode='OBS']/
code[@code='Tnl' and @codeSystemName='Lab Observation Table'
and @displayName='troponin I']/parent::hl7:observation/
hl7:value/attribute::value
```

5.2.4 Data constraints

The CDA schema is based on the XML schema language and can only describe the general structure of the CDA documents. While the CDA schema handles the semantic data

[2] Provided that “*Troponin I*” has a code of “Tnl” in the “Lab Observation Table” vocabulary set.

interoperability by defining semantic placeholders for data items, it does not address the details of the correlations between different elements and constraints on XML attribute values that have to be met. As an example, the encoded data items should have valid ranges. To represent these constraints and check the CDA documents for compliance, we have chosen to represent these application-specific constraints as rules in *schematron documents*.

The schematron language is an XML-based and rule-based language that uses path expressions to refer to data elements within the XML documents. We encode each data item's constraint in the form of an assertion rule in the schematron validating document. At the usage site, the validation engine receives the CDA instance document and validates it with its associated schematron document.

The XML schematron document associated with the data mining model and the CDA instance document further checks for acceptable data values and the interrelations between data pieces and ensures conformance with the input data requirements specification of the mining model. We can also define rules to handle null, missing, or optional values. A sample schematron document that has been developed for our case study is further discussed in Section 5.5.

5.3 Putting things all together

CDA instances are generated according to the adopted CDA schema by the data source owner (e.g., the hospital's EMR system that provides the patient data) and transported to the requester of the data. A messaging framework may have been deployed and used to do the transportation and ensure proper delivery and handling of errors. At the destination site, the receiver receives the XML CDA document. After performing the validation, a parser parses the document to get the encoded data items. Figure 5.2 illustrates this process, and Table 5.1 summarizes the different types of documents

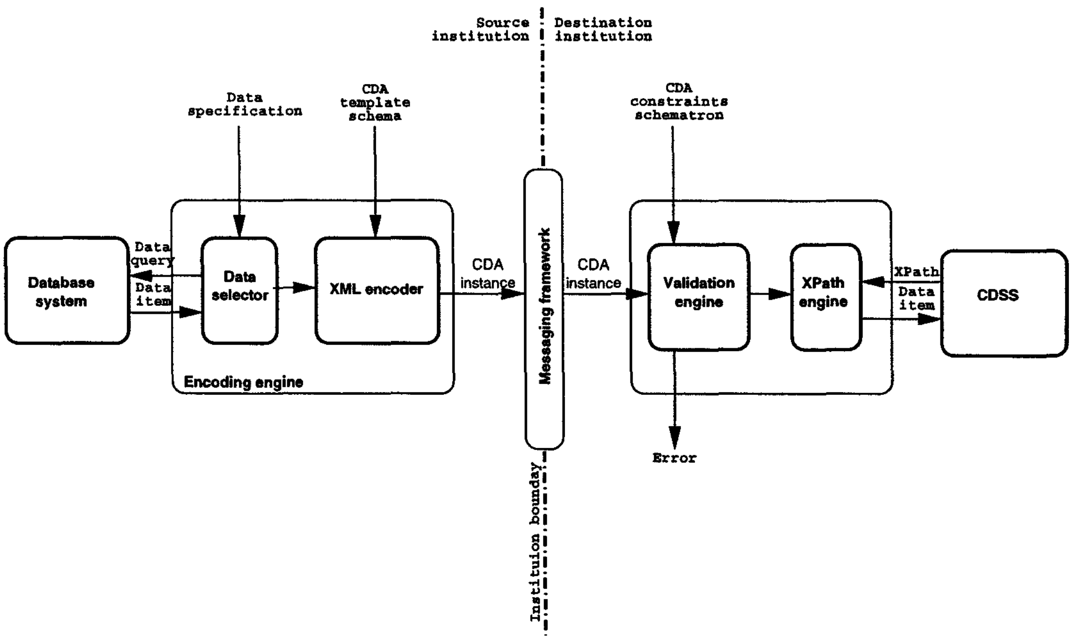

Figure 5.2: Data exchange using CDA.

involved in the exchange of data using CDA.

5.4 Our approach in knowledge interoperability

In the rest of this chapter we elaborate on our approach in knowledge interoperability for both encoding the mined knowledge as well as applying it for the purpose of clinical decision making. We adopt Predictive Model Markup Language (PMML) [21] to encode the mining models that were constructed in the phase 1 of the framework. The PMML specification provides a language to describe various types of mining models, including but not limited to clustering, regression, and association rules models. We use CDA documents to encode the input data, as well as the output results. CDA documents provide data interoperability, while the PMML representation of the knowledge provides knowledge interoperability.

The PMML specification is developed by Data Management Group (DMG) [4] which

Artifact name	Technology	Description
CDA template	XML schema	Describes the structure of the CDA instance.
CDA instance	XML	The XML document containing the encoded data.
Validation document	XML schematron	Describes additional constraints on the data document as imposed by the mining model input requirements.
Data item XPath location	XPath	The XPath address within the CDA instance at which a particular data item is located.

Table 5.1: Different documents and their role in data exchange using CDA.

is an independent and vendor-led group active in developing data mining standards. The specification is in the form of an XML schema or DTD[3] (Document Type Definition). The encoded data mining models will be textual XML documents that conform to this specification. These documents consist of different parts structured according to the schema and specify the model's data types for input and output, necessary transformations of the data before application of the model, and the data mining models themselves.

The encoding process takes place in phase 2 of the framework and is performed by the data analysts, developers, and computer scientists. If the mining model construction was carried out by the use of data mining tools, they may have built-in support and capability to export the results in PMML format. But this activity can also be done manually by the use of conventional XML or text editors. In fact in our case studies that we will present in the Appendix A we have created the model documents manually.

The encoding process should preserve different properties associated with the data mining models. They were chosen by the model builder at the knowledge extraction phase. Below we describe these aspects and how to encode them:

- **Data input/output**: the input data to a data mining model is in the form of a sequence of values (null for missing values). This input sequence has been specified by the data mining performers and should be preserved since in the knowledge interpretation phase of the framework, the data input is provided to the model interpreter in the same format and order. The specification of this sequence (data types) along with the types of the results of the data mining model are encoded in the `<modelSchema>` element.

- **Optional/required or missing values**: optional or required attributes are specified in the PMML models. How to handle missing values should also be specified. Sometimes default values are specified using the `default` attribute of the data ele-

[3] Depending on the version of the specification. Currently, the most recent version, 3.1 is in the form of an XML schema

ments in the PMML specification. These values are substituted for the null values. The choice of the default value is normally the same as the choice that was made when building the model.

- **Input data transformations**: as part of the knowledge discovery process data transformations are performed on the data to make them suitable for the mining. The transformations are normally applied sequentially. For correct interpretation and application of the data mining models, the input data should undergo the same sequence of transformations as the time of mining. For example, consider discretization of the age attribute to bins of *infant*, *youth*, *adult*, and *senior* according to the ranges of $[0, 3)$, $[3, 12)$, $[12, 50)$, and $[50, 120]$. If the age attribute of the mined data set is transformed by this transformation then it should also be performed during the application process too.

- **Data input vocabulary sets and valid value ranges**: a very important issue that should be taken care of is the vocabulary sets that are used in the mining models. During the encoding process we specify categorical attributes' categories (terms) from standard or locally adopted vocabulary sets. These vocabulary sets will also be used in the data encoding process. For the numerical attributes, the valid ranges of values are also specified. The `<dataDictionary>` element of the PMML documents are used to encode such information.

- **Data mining models**: a data structure associated with the data mining model is also encoded. For example, an association rules mining model has a set of 'frequent item sets', each frequent set associated with mining specific parameters, like support and confidence. These parameters should also be preserved, as in some cases they may represent the extent of validity and reliability of the results. For example a high *support* for frequent item sets shows that the extracted association rules were popular in the study data set.

5.4.1 Validating input data items

In the data mining models, the data items are represented using simple and primitive data types (e.g., numerical, categorical, or string types) and there is no notion of complex data structures or application specific data semantics. For this reason, our view of data items just before application of the data mining models is simple and primitive data types that have to conform to pre-specified constraints of each particular mining model's input requirements. These constraints are specified by the builder of the data mining model and encoded as assertion rules in the schematron validation document as described in Section 5.2.4. These rules are checked by the interpretation engine before application of the data mining model.

For example, one type of constraints associated with the input data is the valid range for numerical values and the value set for categorical values. Many clinical data mining studies are often carried out on measured physical data and hence another category of constraints which is implicitly associated with the data inputs is the unit of measurement (associated with the data values in the CDA document). A sample schematron document that has been developed for our case study is further discussed in Section 5.5.

5.4.2 Adding support for new vocabulary and units conversions

Data bindings to different institutions' healthcare data may not provide the data items exactly as they were used and specified in the data mining construction process. For example, the term to refer to a particular medical concept, e.g., a disease, might be different or the measurements be represented in different units. An important and desirable feature for data mining models would then be their support for different vocabulary terms, or different measurement units, for categorical and numerical attributes respectively. To alleviate these inconsistencies and avoid adding new processing components in the application phase, or recoding the whole PMML model with the new vocabularies and data specifications, we define custom transformations to handle them. The XML code snippet below shows a conversion function to transform body temperature represented in Celsius to Fahrenheit.

```
<DataDictionary>
   <DataField name="body-temperature-c" optype="continuous"
      dataType="double"> </DataField>
</DataDictionary>
<TransformationDictionary>
   <DerivedField name="body-temperature-f"
      optype="continuous" dataType="double">
      <NormContinuous field="body-temperature-c">
         <LinearNorm orig="0" norm="32"></LinearNorm>
         <LinearNorm orig="37" norm="99"></LinearNorm>
      </NormContinuous>
   </DerivedField>
</TransformationDictionary>
<MiningModel functionName="...">
   <MiningSchema>
      <MiningField name="body-temperature-f"></MiningField>
   </MiningSchema>
   ...
</MiningModel>
```

5.4.3 Encoding custom results

The PMML specification leaves the way open for developers to define new custom XML structures in the data mining models to encode information that can not be expressed using the general PMML elements and attributes. This has been designed as part of the PMML extendibility feature and its capability to tailor encoded models to incorporate additional information regarding new algorithms. We leverage this feature to annotate our mining models with additional elements containing clinical information appropriate for CDSS. This information can describe the results, e.g., "since the *TDS* index is high, then the result of the melanoma diagnosis is malignant".

More specifically, the mining model encoder annotates the results elements of the data mining model and the corresponding categorical data items in the data dictionary of the PMML document with the additional information that he wants to be accessed by the interpreter and included as part of the results. The custom XML elements are not originally present in the PMML specification and hence additional interpreter support is also required to effectively use them. We use XPath expressions to access this additional application specific information.

Part of this information is encoded in fully structured XML elements that are computer readable. It may provide information about the validity of the results, their support in the mined data, etc.

It is also very helpful to have some human-readable information in natural language as well. This information is represented directly to the user of the CDSS (i.e., the healthcare professional) to gain insight about different aspects of the results. This is particularly important in healthcare, since the user has to be informed of how the resulting output has been achieved. For this purpose, we have adopted to use the CDA text blob specification [30] which provides a rich set of constructs to represent and encode simple ASCII text as well as highly formatted texts. The information may describe the results in more detail, their degree of validity, and how to interpret them. Again note that this information is data mining model- and application-specific. In our case study, we have incorporated some information encoded in natural language text blobs.

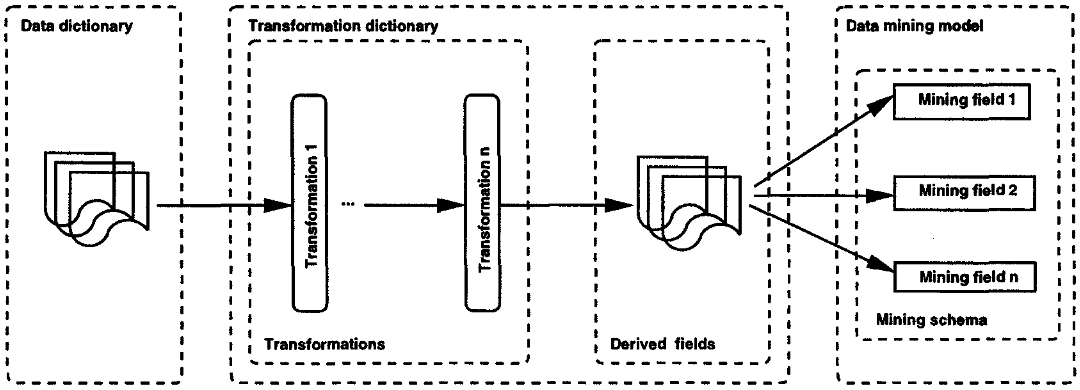

Figure 5.3: Referencing the transformed data items in the mining model's input schema.

5.4.4 Logic modules

To apply the mined knowledge, a program (*logic module*) accesses the required input data items and interprets the data mining model for the patient data by parsing the PMML document that encodes the data mining model. The logic modules provide a simple interface that receives a CDA document as input and generates a CDA document as output.

The input data items are accessed from the CDA document instances which populate the data dictionary. Afterwards, the transformations that are defined in the PMML document are carried out and the derived fields are provided to the encoded data mining model according to its input schemas definition. Figure 5.3 illustrates this process.

Next, the data mining models are applied (interpreted), e.g., a classifier model is applied to classify the data items. After application of the model, intermediate results are produced (e.g., the class which the data items are assigned to). The results are in the form of data structures that are mining model specific, e.g., a set for association rules, or a triggered node in a decision tree. The results are annotated with custom tags that provide additional information about the results. This information has been encoded using the PMML extension mechanism, and is finally placed in a CDA document. Figure 5.4 represents the process within a logic module to apply a data mining model and

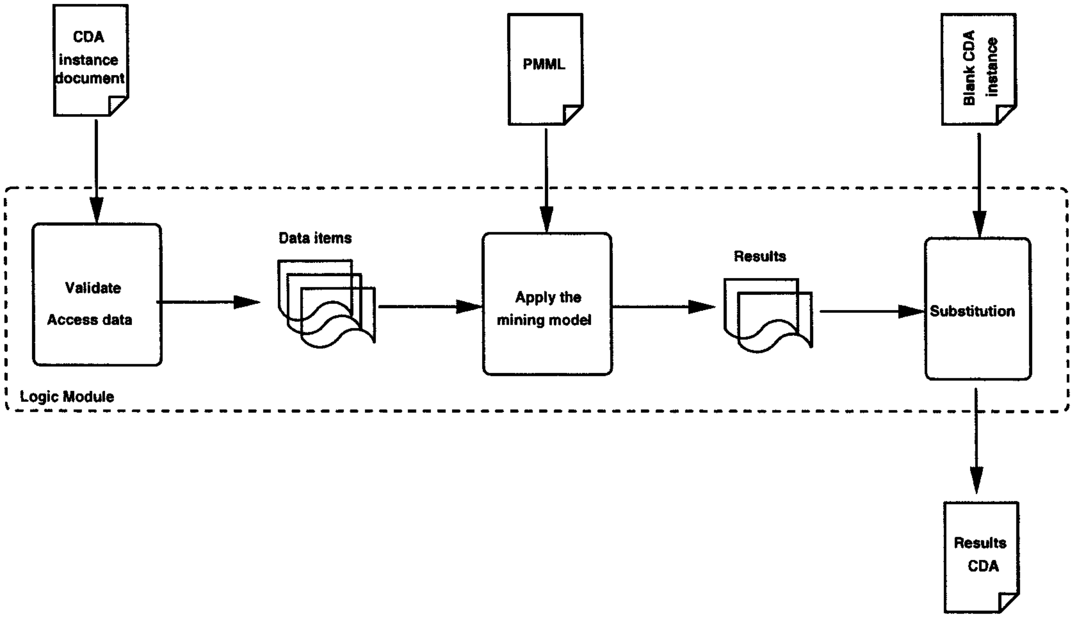

Figure 5.4: The logic modules interprets the mined knowledge for the case data and the results are encoded in the mining model's corresponding CDA document.

generate the results document.

5.5 Running case study

Continuing with our case study from the previous chapter, in this section we provide the details of how to apply the described approach to achieve datâ and knowledge interoperability for this example. The data items for the decision tree model have been identified in Section 4.5 and presented in Table 4.2. The vocabulary set that has been used for encoding the data items and their values is provided in Table A.1.

The data items are encoded using the CDA specification. The XML code snippet below is part of the resulting CDA instance document that encodes the data item for the skin cancer mark's asymmetry using both level 2 and 3 of the CDA specification. The full XML document is provided in Section A.2.

To validate the data items, a schematron document is developed with rules that performs the required checks according to the data items specification which was provided in Table 4.2. The XML code snippet below shows two validation rules. The first rule checks whether the *C-Blue* data item (which is a required field for the classifier) exists in the document or not. The second rule checks the validity of the terms that are used as the value of *C-Blue*.

```
<schema xmlns="http://www.ascc.net/xml/schematron">
    <!--
        Check for existance of ’C-Blue’
    -->
    <pattern name="Check whether the required data elements
    exist.">
        <rule context="/hl7:ClinicalDocument">
            <assert test="
            count(hl7:component/hl7:structuredBody/
            hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
            hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
            hl7:code[@code=’1234-5’ and
            @codeSystem=’2.16.840.1.113883.6.2’])=1">
                The required data element ’C-Blue’ does not
                exist in the input CDA document.
            </assert>
        </rule>
        <!--
            Check for validity of the values of ’C-Blue’
        -->
        <rule context="
        /hl7:ClinicalDocument/hl7:component/
        hl7:structuredBody/
        hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
        hl7:observation[@moodCode='EVN' and @classCode='OBS']/
        hl7:code[@code='1234-5'
        and @codeSystem='2.16.840.1.113883.6.2']">
            <assert test="
            ../hl7:entryRelationship[@typeCode='COMP']/
            hl7:observation[(@classCode='OBS') and
            (@moodCode='EVN')]/hl7:code[
            (@code='1234-5-1' or @code='1234-5-2')]">
                Invalid value for 'C-Blue'
                data element.
            </assert>
        </rule>
    </pattern>
</schema>
```

After validation, the data items are accessed by the logic module from the CDA instance document using XPath expressions. An expression that accesses the value of the *Border* data item in the XML file is provided below. The full list of expressions to access all data items is provided in tables A.2 and A.3 in the appendix.

```
/hl7:ClinicalDocument/hl7:component
/hl7:structuredBody/
hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
hl7:observation[@moodCode='EVN' and @classCode='OBS']/
hl7:code[@code='1234-2' and
@codeSystem='2.16.840.1.113883.6.2']/../hl7:value/attribute::value
```

After the data items are accessed, the logic module calculates the *TDS* index according to equation 4.1. The calculated *TDS* index and the *C-Blue* data item (that is accessed directly from the CDA document) are then input to the data mining model. The data mining model (decision tree) is encoded in a PMML file. The following XML code snippet shows part of the encoded classifier that encodes custom messages that associates the classifier's output along with a description.

```
<DataField displayName="DIAGNOSIS" dataType="string"
    name="DIAG" isCyclic="0" optype="categorical">
    <Value displayValue="Benign-nevus" property="valid"
        value="Benign-nevus" >
        <Extension extender="CAS" name="Description"
            value="TDS is low and C-Blue is absent, so the
            result of classification is Benign-nevus.">
        </Extension>
    </Value>
    <Value displayValue="Blue-nevus" property="valid"
        value="Blue-nevus" >
        <Extension extender="CAS" name="Description"
            value="TDS is low and C-Blue is present, so the
            result of classification is Blue-nevus.">
        </Extension>
    </Value>

    <Value displayValue="Malignant" property="valid"
        value="Malignant" >
        <Extension extender="CAS" name="Description"
```

```
            value="TDS is high, so the result of classification
            is Blue-nevus.">
        </Extension>
    </Value>
    <Value displayValue="Suspicious" property="valid"
        value="Suspicious" >
        <Extension extender="CAS" name="Description"
            value="TDS is in an undecidable range, so the
            result of classification is suspicious.">
        </Extension>
    </Value>

    <Value displayValue="UNKNOWN" property="valid"
        value="UNKNOWN" >
        <Extension extender="CAS" name="Description"
            value="The algorithm can not decide.">
        </Extension>
    </Value>
</DataField>
```

The following code illustrates the structure of the decision tree classifier as encoded in the PMML file. The full PMML code is provided in Section A.5.

```
<TreeModel modelName="Decision Tree Model"
splitCharacteristic="multiSplit" algorithmName="decisionTree"
functionName="classification">
<MiningSchema>
    <MiningField name="TDS" usageType="active" />
    <MiningField name="C-BLUE" usageType="active" />
    <MiningField name="DIAG" usageType="predicted" />
</MiningSchema>

<Node recordCount="245" score="UNKNOWN">
    <True />
    <ScoreDistribution recordCount="62"
        value="Benign-nevus" />
    <ScoreDistribution recordCount="59"
        value="Blue-nevus" />
    <ScoreDistribution recordCount="62"
        value="Malignant" />
    <ScoreDistribution recordCount="62"
        value="Suspicious" />
    <ScoreDistribution recordCount="0"
        value="UNKNOWN" />
    <Node recordCount="121" score="UNKNOWN">
        <SimplePredicate operator="lessOrEqual"
            value="4.85" field="TDS" />
        <ScoreDistribution recordCount="62"
            value="Benign-nevus" />
        <ScoreDistribution recordCount="59"
            value="Blue-nevus" />
        <Node recordCount="62" score="Benign-nevus">
            <SimplePredicate operator="equal"
                value="absent" field="C-BLUE" />
            <ScoreDistribution recordCount="62"
                value="Benign-nevus" />
    </Node>
    <Node recordCount="59" score="Blue-nevus">
        <SimplePredicate operator="equal"
            value="present" field="C-BLUE" />
        <ScoreDistribution recordCount="59"
            value="Benign-nevus" />
    </Node>
</Node>
<Node recordCount="124" score="UNKNOWN">
    <SimplePredicate operator="greaterThan"
        value="4.85" field="TDS" />
    <ScoreDistribution recordCount="62"
        value="Malignant" />
    <ScoreDistribution recordCount="62"
        value="Suspicious" />
    <Node recordCount="62" score="Malignant">
        <SimplePredicate operator="greaterThan"
            value="5.54" field="TDS" />
        <ScoreDistribution recordCount="62"
            value="Malignant" />
    </Node>
    <Node recordCount="62" score="Suspicious">
        <SimplePredicate operator="lessOrEqual"
            value="5.54" field="TDS" />
        <ScoreDistribution recordCount="62"
            value="Suspicious" />
    </Node>
    </Node>
</Node>
</TreeModel>
```

Finally, the results of the application of the data mining model are encoded in a CDA document. The following XML code snippet shows how the result values are encoded. A full sample CDA document for the results is provided in Section A.6.

```
<component>
    <section>
        <code code="1292" codeSystem="2.16.840.1.113883.6.2"
        codeSystemName="CAS Lab Observation Table"
        displayName="Alerts"/>
        <title>Alerts</title>
        <text>
            <table border="1">
                <tbody>
                    <tr>
                        <th>Alert</th>
                        <th>Comments</th>
                        <th>Date</th>
                    </tr>
                    <tr>
                        <td>The result is malignant Melanoma.</td>
                        <td>TDS is high, so the result of
                        classification is Blue-nevus.</td>
                        <td>May 31, 2006</td>
                    </tr>
                </tbody>
            </table>
        </text>
        <entry typeCode="COMP">
            <observation classCode="ALRT" moodCode="INT">
                <code code="2345"
                displayName="Malignant Melanoma"
                codeSystemName="CAS Lab Observation Table"
                codeSystem="2.16.840.1.113883.6.2"></code>
            </observation>
        </entry>
    </section>
</component>
```

Chapter 6

Incorporating mined knowledge in clinical guidelines

We are interested in enhancing the computer-assisted clinical decision making process by leveraging mined clinical knowledge. The knowledge has been extracted in a healthcare data mining analysis process and is represented as a data mining model. The data mining models provide the Clinical Decision Support System with the required decision making logic. For this purpose, a subset of the functionality of a clinical guidelines modeling standard, GLIF3, has been adopted and the language has been extended to incorporate this type of knowledge. By interpreting the mined knowledge for an individual patient's data, the results are presented to the user (i.e., healthcare professional) and the guideline flow is directed accordingly. The knowledge extraction process has been reviewed in chapter 4 and the process of encoding the knowledge along with the data input of the data mining model were described in chapter 5. In this chapter, we will use these contributions in the context of a GLIF3 Clinical Decision Support System.

This chapter first gives an introduction to the Clinical Decision Support Systems in Section 6.1. In Section 6.2 we give a short overview of the Guideline Interchange Format3 (GLIF3) standard. We describe our approach in extending GLIF3 in order to incorporate the mined knowledge in Section 6.3. The implemented guidelines execution environment will be briefly described in Section 6.4.

6.1 Introduction

There is a new paradigm in medical research called *Evidence Based Medicine* (EBM) in which medical researchers identify various approaches in dealing with different clinical problems according to their patients' conditions. They try to collect enough evidence to assess the effectiveness and usefulness of each approach. The practitioners are then encouraged to apply the most effective approach that was proven by various observations and rigorous evaluations.

The dynamic nature of the medical knowledge makes it hard for practitioners to keep up with the pace of changes as well as being able to handle the diversity and the extent of possibilities. Clinical Decision Support Systems are computer programs that are designed to ease the burden on healthcare professionals and help them in providing better care for patients. By providing the clinical best practice knowledge, they help to make evidence based medicine a reality. An immediate desirable outcome of using decision support applications in healthcare would be to speed up the dissemination of clinical best practices, and to reduce the number of medical errors due to human (practitioner) faults by avoiding high reliance on human knowledge.

A Clinical Decision Support System assists the practitioners in the practice of medicine by providing prompts, alerts, and recommendations based on individual patient health data. The best practice medical how-to is encoded and stored within the CDSS application. The CDSS interacts with the practitioner through its user interface, and with the healthcare electronic medical records systems to receive the patient data as input. It will then consult with its knowledge-base and interpret the knowledge therein with regard to the patient's case data at hand. The results of the interpretation is then provided to the

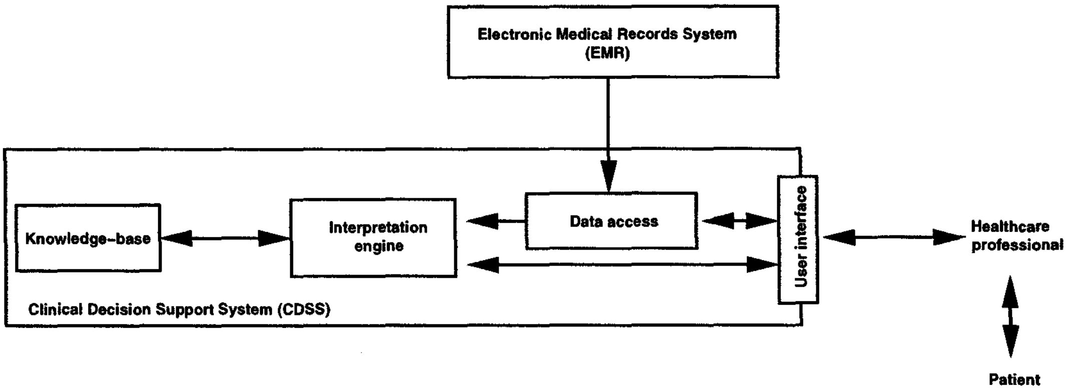

Figure 6.1: The general internal components of a CDSS.

practitioners through the CDSS user interface. Figure 6.1 illustrates different parts of a CDSS.

There are different definitions for Clinical Decision Support Systems in the literature. Below, several of these definitions are presented:

- Clinical Decision Support Systems are systems that access electronic knowledge to help patients, and healthcare providers in making decisions [16].
- Clinical Decision Support Systems are expert systems to aid clinical decision making. They provide assessments or prompts from a knowledge-base which is specific to the patient data [22].
- Clinical Decision Support System are active knowledge systems that generate case-specific advice based on the patient data [32].

In this thesis, we define the Clinical Decision Support Systems as follows:

> *A Clinical Decision Support System is a computer program that interacts with the healthcare professionals and provides them with assistance in the form of timely, and accurate recommendations, alerts, and reminders according to individual patient's medical data.*

The system accesses patient data from electronic patient clinical data repositories, and its knowledge-base that contains credible and current medical knowledge that is provided and endorsed by expert medical researchers. In our research, the knowledge-base contains the results of data mining operations that are encoded as PMML documents and interpreted by the system's *logic module* .

6.2 Guideline modeling language

In this section we describe Guideline Interchange Format 3 (GLIF3) [17], a guideline modeling language that represents the clinical best practices as flow charts. The GLIF specification has been developed by the InterMed Collaboratory, which was a joint project of medical informatics laboratories at Harvard, Stanford, Columbia, and McGill universities. The GLIF guideline models are authored by expert medical researchers according to this specification. They are executed in the Clinical Decision Support System to provide decision making support and clinical best practice how-to to healthcare professionals. GLIF guidelines have been developed for a variety of purposes, including but not limited to heart failure, hypertension, thyroid screening, and many more.

GLIF3 guidelines are defined in three levels of abstraction:

- **Conceptual level**: the first level is a flow chart that represents different states and actions in a structured graph. This level provides an easy to comprehend conceptualization of the guideline. At this level, the details of decision making are not provided and hence the guideline models are not computable.

- **Computable level**: to make the guideline flows computable, the author has to specify the control flow, decision criteria, medical concepts, and relevant patient data. These are specified in the computable level.

- **Implementation level**: for GLIF guidelines to be actually deployed at an institution site, the patient data and actions should be mapped to institution specific information systems. The required mappings are specified in this level[1].

Figure 6.2 illustrates a GLIF3 guideline at the first level of abstraction. Five different types of steps (nodes in the flow graph) are present in the conceptual level:

- *Decision step* is a node in a guideline model's flow graph that determines the direction of the flow based on a decision criterion specified in an expression language. For example, the age of the patient might be compared to a specific age as a decision criterion to direct the flow.

- *Activity step* is a node that performs an action, e.g., prompts to prescribe medications; order tests; retrieve patient's medical records; or recommends treatments.

- *Patient state step* is a node in the flow graph that designates a specific patient's condition, e.g., presence of a symptom, previous treatments, or diagnoses. Also, guideline models start with a patient state step.

- *Branch step* is used to fork and generate two or more concurrent decision making guideline-flows; such as, ordering a lab test and prescribing medication both at the same time.

- *Synchronization step* is used to merge two or more concurrent decision flows into a single decision flow; such as, receiving the lab test report, and observing the effectiveness of the prescribed medication, before continuing to proceed to the next step.

[1] The specification of these details is not yet completed in the GLIF specification.

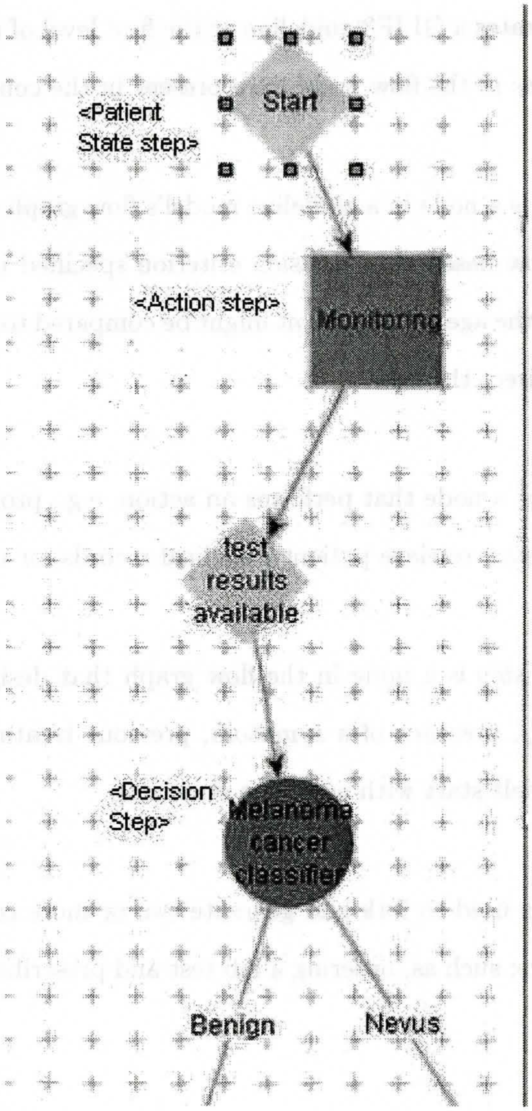

Figure 6.2: A clinical guideline model in the first level of abstraction.

6.3 Decision making based on the results of data mining analysis

To incorporate data mining extracted knowledge for clinical decision making, we have adopted the GLIF3 specification at the conceptual level and extended it. At the computable level we have provided our own approach to handle data and decision making logic which is based on the data and knowledge interoperability approaches that were discussed in chapter 5. The details of the implementation level are discussed in the next section.

The reason for choosing GLIF3 was the simplicity and understandability of its models by using graphical representations for clinical guidelines. This is an important factor, since the medical researchers who develop and are familiar with the clinical how-to are often not computer experts. Hence, it is desirable to avoid complexity in defining the models. Also, the practitioners who use the system would like to know and understand the underlying guidelines. This helps the system to achieve better acceptance. However, despite the simplicity, the GLIF3 models have the necessary constructs and modeling elements to define very sophisticated and complex guidelines. In short, GLIF3 provides the required amount of expressiveness that we need as a base to add our data mining based decision making logic on top.

6.3.1 Conceptual level

At the conceptual level, the GLIF3 modeling constructs are represented as ontology classes[2]. We have defined a new abstract class, *data mining entity* with a slot (attribute) *logic_module*. This slot holds the name of the logic module that is used for interpreting an associated data mining model. Two new classes, called *data mining decision step* and *data mining patient step* are then defined that extend the *data mining entity* class,

[2] An ontology is a data model that represents the concepts within a domain.

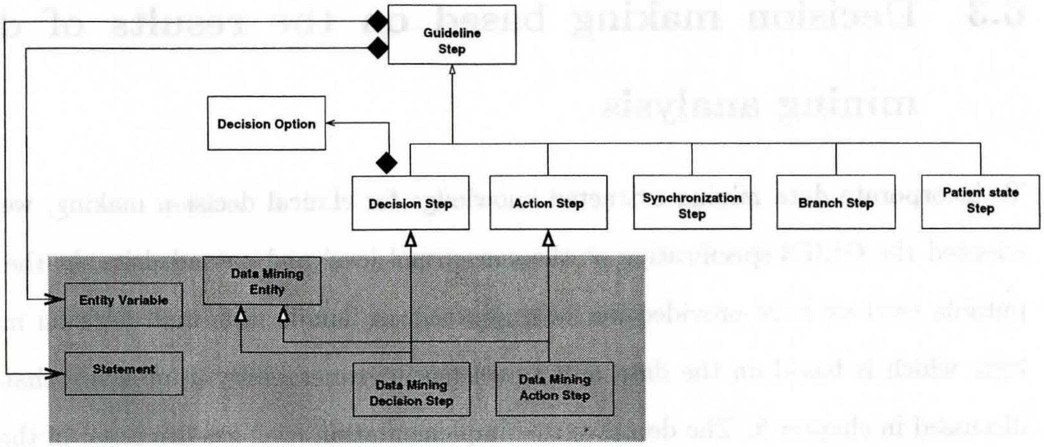

Figure 6.3: Top-level view of the modified GLIF3 meta model. The proposed extension classes are shown in the shaded area.

as well as the *decision step* and *patient step* classes respectively. These classes add the functionality that is necessary to access and interpret a data mining model. Figure 6.3 illustrates the meta model diagram of the extended GLIF.

Guideline models are represented as instances of the *algorithm* ontology class (not shown in the diagram), and different steps are instantiated from corresponding classes. Also, the top level class, *guideline step* (and hence its sub-classes), contains a *statement* class that is used at the computable level to invoke pre-defined functions that manipulate data and variables, to apply the mined knowledge from the knowledge-base, or output the results to the user. These functions are invoked by the execution engine, whenever the flow arrives to that step.

6.3.2 Computable level

At the computable level, we provide the functionality to manipulate variables [3] within the guideline steps by invoking functions. Using the *statement* class in the *guideline step*

[3] Variables are not explicitly declared

class, a set of variables can be defined within each step. The variable names consist of characters without spaces; and to reference the value of a variable, its name is prefixed with a $. The scope of the variables is limited to a single step. The functions that are defined are as follows:

- *getDocument(String varName, String location, String docID, String patientID)*:

  This function retrieves a CDA document from the location that is specified by *location*. The third parameter, *docID*, specifies the type of CDA document to be retrieved, and the last argument, *patientID*, specifies a patient identifier who the CDA document belongs to. After the document is retrieved its content is read into the variable specified by *varName*.

- *getDataItem(String varName, String cdaDoc, String xpath)*:

  This function assigns a value from the location specified with *xpath* from the CDA document *cdaDoc*, to the variable specified by *varName* with the data item in a CDA document. The CDA document's content is specified in the *cdaDoc* argument and the XPath expression referring to the data item in the CDA document is specified by the *xpath* argument.

- *setVariable(String varName, String value)*:

  This function assigns the string value specified by *value* to the variable name specified by *varName*. Variable names contained in *value* are replaced by their associated values.

- *evaluateExpression(String varName, String expr)*:

  This function evaluates the logical expression specified by *expr* and assigns the resulting value to the variable *varName*.

- *logicModule(String cdaResult, String cdaInput)*:

  This function invokes the logic module specified in the step's *logic_module* slot with the CDA document's content in *cdaInput.* After the module is invoked, the output which is in the form of a CDA document is stored in the variable specified by *cdaResult.* This function is only available at the *data mining decision step* and *data mining patient step* s. Having the results in the form of a CDA document has this advantage to use the same XPath referencing mechanism (provided by the *getDataItem* function) to access different result data items.

- *alert(String message):*

  This function outputs the message specified by *message* to the user. The message can contain a patient-specific alert, recommendation, or reminder. Variable names contained in *message* are substituted by their associated values.

As the flow arrives at a step, the functions specified in that step are invoked. Based on the type of the step, related actions take place. For example at *decision steps* different options (the following steps) are presented to the user. This enables the user to make the final decision and decide which path the flow should go to.

6.4 The guideline execution environment

There are basically two approaches in executing a guideline model [43]. In the first approach, new software is built for individual guideline instances that implement the guideline flow as specified by the interconnection of the steps. This approach has many drawbacks as time is wasted re-developing much of the functionalities. Also, small changes in the model may require considerable recoding. Hence, necessary flexibility is obviously missing and therefore, this is not considered as a very favorable approach.

On the other hand, we can think of an environment with an engine that receives a guideline model as input and interprets the model according to its specification. In this environment, we define a set of software modules that are responsible to performing the necessary actions as determined by the guideline model. The environment's execution engine is capable of following the guideline model and invoking the corresponding modules. During the execution of the guideline, the environment keeps track of the guideline's execution flow and provides the required data retrieval and knowledge interpretation facilities.

We adopted the second approach and implemented an environment and its execution engine to automatically interpret and execute a clinical guideline that has been defined according to the specification that we described in the previous section.

In our implementation of the guidelines execution environment, we have developed a plug-in in the *Protégé* [3] ontology editor tool. GLIF modeling constructs are represented as ontology classes and guideline authoring is done in a graph widget in *Protégé* using the classes that we defined in the previous section. Figure 6.4 illustrates a snapshot of the execution environment.

The environment allows multiple guideline models to be defined and the user can select a guideline for execution from a list. An instance of the engine is then instantiated to execute the selected guideline model. Execution is started from the initial step and continues along the links that connect different steps. The engine supports multiple flows of execution for each running guideline, since individual flows can fork at *branch steps*.

Each flow refers to a guideline step as its active step. Active steps are executed by the engine upon arrival of the flow to that step. After execution of a step and providing the user with the outputs, the engine waits for the user to signal continuation of each flow to the next step. At this point, the engine retrieves the next step from the ontology model, updates the signaled flow, and executes the new active step.

The logic modules are implemented in Java and are run wherever the *logicModule* function is invoked within a *data mining decision step* or *data mining patient step* in the guideline model. They access locally stored PMML files that contain the corresponding data mining model. We implemented them by using the XELOPES library [13]. As described in chapter 5 the outputs of the modules are CDA documents. In our imple-

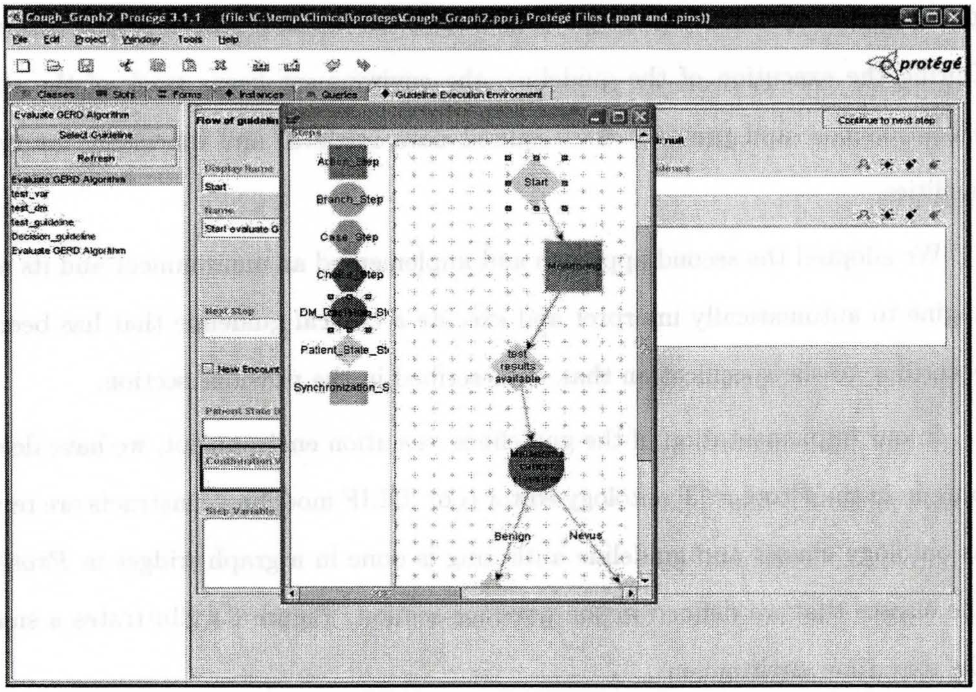

Figure 6.4: The guideline execution environment within the *Protégé* ontology editor.

mentation, to run a *logic module* a Java class is loaded. The class implements the Java interface *LogicModule* which has a single method:

public String runLogicModule(String inputCDA);

The CDA document is passed as a *String* input parameter to the *runLogicModule* function which accesses the required data items. After reading and parsing the associated PMML document, it applies the data mining model. Finally, the output results are encoded as a CDA document and is stored in a variable in the invoking guideline step.

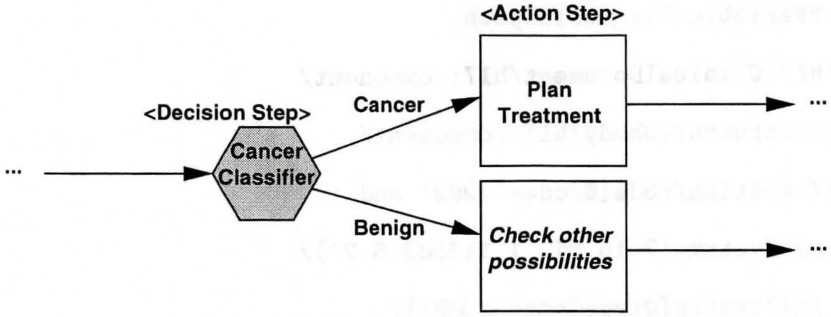

Figure 6.5: A portion of the guideline model for the Melanoma skin cancer classifier.

6.5 Running case study

Continuing with our case study from the previous chapters, in this section we demonstrate parts of a guideline model that was developed for the Melanoma skin cancer decision tree classifier. The guideline model at its conceptual level is illustrated in figure 6.5.

At the *data mining decision step* the following functions are encoded to access and retrieve the required data items as well as running the *logic module* and providing the results and available options to the user.

1. Read the content of the CDA document containing the required data items for the data mining model into `cdaVar` variable.
   `getDocument(cdaVar, "/data-repository", "skin-tests", "Reza Sherafat")`

2. Execute the logic module to interpret the data mining model for the patient data and store the resulting CDA document in `resultCdaVar`.
   `logicModule(resultCdaVar, $cdaVar)`

3. Define the variable to hold the XPath expressions referencing the result of the classification from the output CDA document.

```
setVariable(DiagnosisXpath,
"/hl7:ClinicalDocument/hl7:component/
hl7:structuredBody/hl7:component/
hl7:section/code[@code='1292' and
@codeSystem='2.16.840.1.113883.6.2']/
../hl7:entry[@typeCode='COMP']/
observation[@classCode='ALRT'
and @moodCode='INT']/code/attribute::code")

setVariable(DiagnosisDescriptionXpath,
"/hl7:ClinicalDocument/hl7:component/
hl7:structuredBody/hl7:component/
hl7:section/code[@code='1292' and
@codeSystem='2.16.840.1.113883.6.2']/
../hl7:text")
```

4. Retreieve the classification result from the resulting CDA document.

```
getDataItem(resultVar1, $resultCdaVar, $DiagnosisXpath)
getDataItem(resultVar2, $resultCdaVar, $DiagnosisDescriptionXpath)
```

5. Output a message to the user with the content of the classification.

```
alert("The result of the Melanoma skin cancer classifier has been
$resultsVar1. Descriptions follow. Please select an option to continue:
$resultsVar2")
```

Chapter 7

Conclusion

In this thesis, we described a novel framework for dissemination and application of the data and mined knowledge among heterogeneous healthcare information systems. For data interoperability, we used CDA schema to define the structure for encoding patients' health related data. We used schematron documents to define validation rules that perform consistency checks on CDA document instances. The healthcare researchers extract data mining results by mining healthcare data. We further used the PMML specification, to encode produced mined knowledge to achieve knowledge interoperability between sources of knowledge and their users. To our best knowledge, there has been no method or technique prior to this research to make this type of knowledge available at the application sites. In the proposed framework, logic modules will access patient data from CDA documents and data mining models from the PMML documents, and interpret that. For futher interoperability, the results of this interpretation are also provided as CDA documents. Furthermore, we applied our framework in a clinical guidelines modeling language, GLIF3. For this purpose, we chose a simplified subset of GLIF3 and extended it. The Clinical Decision Support Systems that are developed based on the extended guideline models provide healthcare professionals with the decision making logic that has been extracted in a data mining analysis. We demonstrated the application of our framework on a case study from the literature. Finally, we descibed a prototype tool that has been implemented.

However, we note that there are still many obstacles that have to be taken care of. In the remainder of this chapter we provide a short list of the most significant ones.

Before a Clinical Decision Support System can be deployed in a real world healthcare environment, it must undergo rigorous evaluations and tests to ensure safety and high quality of the decision making. The knowledge-base content, the system, and its processes need careful attention. Due to the fact that automatic techniques are used for knowledge extraction in data mining operations from large healthcare data repositories, it is usually hard and time consuming to perform the evaluation.

Due to the liability issues that the healthcare practitioners are concerned with, Clinical Decision Support Systems perform a critical task. Also, adaptation by healthcare professionals requires more time, and all system outputs should be examined carefully by the user. Short times allocated to a single patient visit are quite common in today's medical practice, and as a result in many cases the practitioners are sensitive about the interaction time with the system.

The approach taken in this research represents the mined knowledge as PMML models which are self-describing XML documents. The research should continue to examine the incorporation of the results of different types of data mining techniques. Fortunately, the database and data mining communities have been active in recent years to define and extend the PMML specification to a variety of different types of models, as well as developing libraries to implement different algorithms. Our research is focused on an application area of data mining results in the healthcare domain rather than the knowledge extraction process. We hope that ongoing research in the data mining area provides widely available implementations of the PMML specification for researchers to apply and expand our approach in mined knowledge dissemination and application into new domains.

Appendix A

A case of classification models

In this section we consider the Melanoma skin cancer as our case study and provide the complete source of different XML documents that were developed as part of the data and knowledge interoperability process. In the previous chapters, different segments of these documents were discussed.

A.1 Vocabulary set

Table A.1 on page 94 presents the vocabulary set that has been used for encoding data items and their values for the Melanoma skin cancer classifier in the generated CDA documents. These codes and names are defined in the "CAS Lab Observation Table" vocabulary system. This vocabulary set is identified by the code "2.16.840.1.113883.6.2".

A.2 CDA instance document

The code below illustrates a sample CDA instance document that encodes the data items required for the Melanoma skin cancer classifier. The CDA schema that describes the structure of this XML document has been adopted from the *e*-MS project.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Display name	Code
Asymmetry	1234-1
Symmetric-spot	1234-1-0
1-axial asymmetry	1234-1-1
2-axial asymmetry	1234-1-2
Border	1234-2
Color	1234-3
White	1234-3-1
Blue	1234-3-2
Black	1234-3-3
Red	1234-3-4
Light brown	1234-3-5
Dark brown	1234-3-6
Diversity	1234-4
Pigment globules	1234-4-1
Pigment dots	1234-4-2
Branched strikes	1234-4-3
Structureless areas	1234-4-4
Pigment network	1234-4-5
C-Blue	1234-5
Present	1234-5-1
Absent	1234-4-2
Benign Melanoma	2348
Blue Melanoma	2347
Suspicious	2346
Malignant Melanoma	2345

Table A.1: The vocabulary terms and their corresponding codes in the "CAS Lab Observation Table" vocabulary set.

```
<ClinicalDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:hl7-org:v3 ../EMS/EMS_CDA.xsd"
 xmlns="urn:hl7-org:v3">
<!--
    ************
    CDA Header
    ************
-->
    <id extension="6823AC2F-B374-4214-AF1B-407091BBED37"
    root="2.16.840.1.113883.3.933"/>
    <code code="34140-4" codeSystem="2.16.840.1.113883.6.1"
    codeSystemName="LOINC" displayName="Referral"/>
    <title>Referral Letter for Melanoma skin cancer diagnosis.</title>
    <effectiveTime value="20060531121533"/>
    <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
<!--
    ************
    Information Recipient
    ************
-->
    <informationRecipient typeCode="PRCP">
        <intendedRecipient classCode="ASSIGNED">
            <id extension="ksartipi"
            root="DCCD2C68-389B-44c4-AD99-B8FB2DAD1493"/>
            <informationRecipient>
                <name>
                    <prefix>Dr.</prefix>
                    <given>Kamran</given>
                    <given></given>
                    <family>Sartipi</family>
                    <suffix></suffix>
                </name>
            </informationRecipient>
        </intendedRecipient>
    </informationRecipient>
<!--
    ************
    Author
    ************
-->
    <author>
        <time value="20060612120000"/>
        <assignedAuthor>
            <id extension="hhippocrates"
            root="DCCD2C68-389B-44c4-AD99-B8FB2DAD1493"/>
        </assignedAuthor>
    </author>
    <author>
        <time value="20040812120000"/>
        <assignedAuthor>
            <id extension="hhippocrates"
            root="DCCD2C68-389B-44c4-AD99-B8FB2DAD1493"/>
            <assignedAuthoringDevice>
                <softwareName>oXygen XML</softwareName>
            </assignedAuthoringDevice>
        </assignedAuthor>
    </author>
<!--
    ************
    Custodian
    ************
-->
    <custodian typeCode="CST">
        <assignedCustodian>
            <representedCustodianOrganization>
                <id extension="888"
                root="7EEF0BCC-F03E-4742-A736-8BAC57180C5F"/>
                <name>CAS at McMaster University</name>
            </representedCustodianOrganization>
        </assignedCustodian>
    </custodian>
<!--
    ************
    Record Target for patient "Reza Sherafat"
    ************
-->
    <recordTarget typeCode="RCT" contextControlCode="OP">
        <patient classCode="PAT">
            <id extension="9999999999"
            root="2BFBA1E9-79C2-4bbb-B589-41B949BD6A3B"
                assigningAuthorityName="PHN"/>
            <patientPatient>
                <name>
                    <prefix>Mr.</prefix>
                    <given>Reza</given>
                    <given></given>
                    <family>Sherafat</family>
                    <suffix></suffix>
                </name>
                <administrativeGenderCode code="M"
                codeSystem="2.16.840.1.113883.5.1"
                codeSystemName="HL7 - Administrative Gender"
                displayName="Male"/>
                <birthTime value="19820622"/>
            </patientPatient>
        </patient>
    </recordTarget>
<!--
    ************
    CDA Structured Body
    ************
-->
    <component>
        <structuredBody>
<!--
    ************
    Purpose
    ************
```

```
-->
            <component>
                <section>
                    <code code="001"
                    codeSystem="7BA9BFFD-D25F-44e8-A7B0-0DF214D6845B"
                    codeSystemName="e-MS Document Section Codes"
                    displayName="Purpose"/>
                    <title>Purpose Section</title>
                    <text>
                        <paragraph>
                            Diagnosis of Melanoma skin cancer
                        </paragraph>
                        <paragraph>
                            Urgency: <content styleCode="bold">
                            Expedite (Call)</content>
                        </paragraph>
                    </text>
                </section>
            </component>
<!--
    ************
    Labs
    ************
-->
            <component>
                <section>
                    <code code="11502-2"
```

```
codeSystem="2.16.840.1.113883.6.1"
codeSystemName="LOINC" displayName="Labs"/>
<title>Labs</title>
<text>
    <table border="1">
        <tbody>
            <tr>
                <th>Type</th>
                <th>Collection Date</th>
                <th>Result</th>
            </tr>
            <tr>
                <td>Skin cancer's mark asymmetry</td>
                <td>May 31, 2006</td>
                <td>1-axial asymmetry</td>
            </tr>
            <tr>
                <td>Skin cancer's mark border</td>
                <td>May 31, 2006</td>
                <td>4</td>
            </tr>
            <tr>
                <td>Skin cancer's mark color</td>
                <td>May 31, 2006</td>
                <td>white, blue, red</td>
            </tr>
            <tr>
```

```
                <td>Skin cancer's mark diversity</td>
                <td>May 31, 2006</td>
                <td>structureless areas, pigment
                network</td>
            </tr>
            <tr>
                <td>C-Blue</td>
                <td>May 31, 2006</td>
                <td>present</td>
            </tr>
        </tbody>
    </table>
</text>
<entry typeCode="COMP">
    <observation moodCode="EVN" classCode="OBS">
        <code code="1234-1"
        codeSystemName="CAS Lab Observation Table"
        codeSystem="2.16.840.1.113883.6.2"
        displayName="Skin cancer's mark asymmetry">
        </code>
        <effectiveTime value="20060531"/>
        <entryRelationship typeCode="COMP">
            <observation classCode="OBS"
            moodCode="EVN">
                <code code="1234-1-1"
                displayName="1-axial asymmetry"
                codeSystem="2.16.840.1.113883.6.2"
                codeSystemName=
                "CAS Lab Observation Table">
                </code>
            </observation>
        </entryRelationship>
    </observation>
</entry>


<entry typeCode="COMP">
    <observation moodCode="EVN" classCode="OBS">
        <code code="1234-2"
        codeSystemName="CAS Lab Observation Table"
        codeSystem="2.16.840.1.113883.6.2"
        displayName="Skin cancer's mark border">
        </code>
        <effectiveTime value="20060531"/>
        <value xsi:type="PQ" value="4">
        </value>
    </observation>
</entry>
<entry typeCode="COMP">
    <observation moodCode="EVN" classCode="OBS">
        <code code="1234-3"
        codeSystemName="CAS Lab Observation Table"
        codeSystem="2.16.840.1.113883.6.2"
        displayName="Skin cancer's mark color">
        </code>
```

```
<effectiveTime value="20060531"/>
<entryRelationship typeCode="COMP">
    <observation classCode="OBS"
    moodCode="EVN">
        <code code="1234-3-1"
        displayName="white"
        codeSystem="2.16.840.1.113883.6.2"
        codeSystemName=
        "CAS Lab Observation Table">
        </code>
    </observation>
</entryRelationship>
<entryRelationship typeCode="COMP">
    <observation classCode="OBS" moodCode="EVN">
        <code code="1234-3-2" displayName="blue"
        codeSystem="2.16.840.1.113883.6.2"
        codeSystemName=
        "CAS Lab Observation Table">
        </code>
    </observation>
</entryRelationship>
<entryRelationship typeCode="COMP">
    <observation classCode="OBS" moodCode="EVN">
        <code code="1234-3-4" displayName="red"
            codeSystem="2.16.840.1.113883.6.2"
            codeSystemName=
            "CAS Lab Observation Table">
```

```
                </code>
            </observation>
        </entryRelationship>
    </observation>
</entry>
<entry typeCode="COMP">
    <observation moodCode="EVN" classCode="OBS">
        <code code="1234-4"
        codeSystemName="CAS Lab Observation Table"
        codeSystem="2.16.840.1.113883.6.2"
        displayName="Skin cancer's mark diversity">
        </code>
        <effectiveTime value="20060531"/>
        <entryRelationship typeCode="COMP">
            <observation classCode="OBS" moodCode="EVN">
                <code code="1234-4-4"
                displayName="structureless areas"
                codeSystem="2.16.840.1.113883.6.2"
                codeSystemName=
                "CAS Lab Observation Table">
                </code>
            </observation>
        </entryRelationship>
        <entryRelationship typeCode="COMP">
            <observation classCode="OBS" moodCode="EVN">
                <code code="1234-4-5"
                displayName="pigment network"
```

```
                        codeSystem="2.16.840.1.113883.6.2"
                        codeSystemName=
                        "CAS Lab Observation Table">
                        </code>
                    </observation>
                </entryRelationship>
            </observation>
        </entry>
        <entry typeCode="COMP">
            <observation moodCode="EVN" classCode="OBS">
                <code code="1234-5"
                codeSystemName="CAS Lab Observation Table"
                codeSystem="2.16.840.1.113883.6.2"
                displayName="C-Blue"></code>
                <effectiveTime value="20060531"/>
                <entryRelationship typeCode="COMP">
                    <observation classCode="OBS" moodCode="EVN">
                        <code code="1234-5-1" displayName="present"
                        codeSystem="2.16.840.1.113883.6.2"
                        codeSystemName="CAS Lab Observation Table">
                        </code>
                    </observation>
                </entryRelationship>
            </observation>
        </entry>
    </section>
</component>
```

```
        </structuredBody>
    </component>
</ClinicalDocument>
```

A.3 CDA validation document

The code below is the schematron document to validate the CDA document instances that contain the data items of the data mining model for classification of Melanoma skin cancer.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--
    This is the schematron document to validate the data values
    in the Melanoma skin cancer referral CDA document.
-->
<schema xmlns="http://www.ascc.net/xml/schematron"
    xmlns:sch="http://www.ascc.net/xml/schematron"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.ascc.net/xml/schematron
    http://www.ascc.net/xml/schematron/schematron1-5.xsd"
    xmlns:hl7="urn:hl7-org:v3">
    <title>
        Melanoma skin cancer referral CDA schematron rule
        definitions
    </title>
    <ns uri="urn:hl7-org:v3" prefix="hl7"/>
    <ns uri="http://www.w3.org/2001/XMLSchema-instance"
    prefix="xsi"/>
```

```
<!--
    Check the existance of required data elements.
-->
<pattern name="Check whether the required data elements
exist.">
    <rule context="/hl7:ClinicalDocument">
        <assert test="
        count(hl7:component/hl7:structuredBody/
        hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
        hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
        hl7:code[@code=’1234-5’ and
        @codeSystem=’2.16.840.1.113883.6.2’])=1">
            The required data element ’C-Blue’ does not
            exist in the input CDA document.
        </assert>
    </rule>
    <rule context="/hl7:ClinicalDocument">
        <assert test="
        count(hl7:component/hl7:structuredBody/
        hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
        hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
        hl7:code[@code=’1234-1’
        and @codeSystem=’2.16.840.1.113883.6.2’])=1">
            The required data element ’Skin cancer’s mark
            asymmetry’ does not exist in the input CDA
            document.
    </assert>
</rule>
<rule context="/hl7:ClinicalDocument">
    <assert test="
    count(hl7:component/hl7:structuredBody/
    hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
    hl7:observation[@moodCode='EVN' and @classCode='OBS']/
    hl7:code[@code='1234-2'
    and @codeSystem='2.16.840.1.113883.6.2'])=1">
        The required data element 'Skin cancer's mark
        border' does not exist in the input CDA document.
    </assert>
</rule>
<rule context="/hl7:ClinicalDocument">
    <assert test="
    count(hl7:component/hl7:structuredBody/
    hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
    hl7:observation[@moodCode='EVN' and @classCode='OBS']/
    hl7:code[@code='1234-3'
    and @codeSystem='2.16.840.1.113883.6.2'])=1">
        The required data element 'Skin cancer's mark
        color' does not exist in the input CDA document.
    </assert>
</rule>
<rule context="/hl7:ClinicalDocument">
    <assert test="
    count(hl7:component/hl7:structuredBody/
        hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
        hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
        hl7:code[@code=’1234-4’
        and @codeSystem=’2.16.840.1.113883.6.2’])=1">
            The required data element ’Skin cancer’s mark
            diversity’ does not exist in the input CDA document.
        </assert>
    </rule>
</pattern>
<pattern
name="Check whether the data elements have valid ranges/values.">
    <rule context="
    /hl7:ClinicalDocument/hl7:component/hl7:structuredBody/
    hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
    hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
    hl7:code[@code=’1234-1’
    and @codeSystem=’2.16.840.1.113883.6.2’]">
        <assert test="
        ../hl7:entryRelationship[@typeCode=’COMP’]/
        hl7:observation[(@classCode=’OBS’)
        and (@moodCode=’EVN’)]/hl7:code[(@code=’1234-1-0’
        or @code=’1234-1-1’ or @code=’1234-1-2’)
        and @codeSystem=’2.16.840.1.113883.6.2’]">
            Invalid value for Skin cancer’s mark ’asymmetry’
            data element.
        </assert>
    </rule>
```

```
<rule context="
/hl7:ClinicalDocument/hl7:component
/hl7:structuredBody/
hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
hl7:code[@code=’1234-2’
and @codeSystem=’2.16.840.1.113883.6.2’]">
    <assert test="../hl7:value[(@xsi:type=’PQ’) and
        (number(@value)>=0) and (number(@value)&lt;=8)]">
        Invalid value for ’Skin cancer’s mark border’
        data element.
    </assert>
</rule>
<rule context="
/hl7:ClinicalDocument/hl7:component/
hl7:structuredBody/
hl7:component/hl7:section/hl7:entry[@typeCode=’COMP’]/
hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/
hl7:code[@code=’1234-3’
and @codeSystem=’2.16.840.1.113883.6.2’]
">
    <assert test="
    ../hl7:entryRelationship[@typeCode=’COMP’]/
    hl7:observation[(@classCode=’OBS’)
    and (@moodCode=’EVN’)]/hl7:code[@code=’1234-3-1’ or
    @code=’1234-3-2’ or @code=’1234-3-3’ or
    @code=’1234-3-4’ or @code=’1234-3-5’ or
```

```
        @code='1234-3-6']">
            Invalid value for 'Skin cancer's mark color'
            data element.
        </assert>
    </rule>
    <rule context="
    /hl7:ClinicalDocument/hl7:component/
    hl7:structuredBody/
    hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
    hl7:observation[@moodCode='EVN' and @classCode='OBS']/
    hl7:code[@code='1234-4' and @codeSystem='2.16.840.1.113883.6.2']/
    hl7:entryRelationship[@typeCode='COMP']/
    hl7:observation[@classCode='OBS' and @moodCode='EVN']/hl7:code
    ">
        <assert test="
            @code='1234-4-1' or @code='1234-4-2' or
            @code='1234-4-3' or
            @code='1234-4-4' or @code='1234-4-5'">
            Invalid value for 'Skin cancer's mark diversity'
            data element.
        </assert>
    </rule>
    <rule context="
    /hl7:ClinicalDocument/hl7:component/
    hl7:structuredBody/
    hl7:component/hl7:section/hl7:entry[@typeCode='COMP']/
    hl7:observation[@moodCode='EVN' and @classCode='OBS']/
```

```
        hl7:code[@code='1234-5'
        and @codeSystem='2.16.840.1.113883.6.2']">
            <assert test="
            ../hl7:entryRelationship[@typeCode='COMP']/
            hl7:observation[(@classCode='OBS') and
            (@moodCode='EVN')]/hl7:code[
            (@code='1234-5-1' or @code='1234-5-2')]">
                Invalid value for 'C-Blue'
                data element.
            </assert>
        </rule>
    </pattern>
</schema>
```

A.4 XPath expressions to access data items from the input CDA instance document

Tables A.2 and A.3 lists the XPath expressions used to access the data items for the Melanoma skin cancer case study.

A.5 Data mining model

The following code is the XML document for the PMML model that encodes the Melanoma skin cancer classifier.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
/*********************************************************/
```

Data item	XPath
Skin cancer’s mark asymmetry	/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/hl7:component/ hl7:section/hl7:entry[@typeCode=’COMP’]/ hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/ hl7:code[@code=’1234-1’ and @codeSystem=’2.16.840.1.113883.6.2’]/ ../hl7:entryRelationship [@typeCode=’COMP’]/hl7:observation [(@classCode=’OBS’) and (@moodCode=’EVN’)]/hl7:code/ attribute::code
Skin cancer’s mark border	/hl7:ClinicalDocument/hl7:component /hl7:structuredBody/ hl7:component/hl7:section/ hl7:entry[@typeCode=’COMP’]/ hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/ hl7:code[@code=’1234-2’ and @codeSystem=’2.16.840.1.113883.6.2’]/ ../hl7:value/attribute::value
Skin cancer’s mark color	/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/ hl7:component/hl7:section/ hl7:entry[@typeCode=’COMP’]/ hl7:observation[@moodCode=’EVN’ and @classCode=’OBS’]/ hl7:code[@code=’1234-3’ and @codeSystem=’2.16.840.1.113883.6.2’]/ ../hl7:entryRelationship [@typeCode=’COMP’]/ hl7:observation[(@classCode=’OBS’) and (@moodCode=’EVN’)]/ hl7:code/attribute::code

Table A.2: XPath expressions used to access the data items for the Melanoma skin cancer classifier.

```
/*        THIS IS THE CLASSIFICATION MODEL FOR           */
/*             DIAGNOSIS OF MELANOMA SKIN                */
/*    CANCER BASED ON THE ALGORITHM PRESENTED IN:        */
/*                       xxx                             */
/*                                                       */
/*        http://www.cas.mcmaster.ca/~sherafr            */
/*                                                       */
/*              Created on May 27, 2006                  */
/*                                                       */
/*                                                       */
/*         Written by: Reza SHERAFAT KAZEMZADEH          */
/*     In Partial Fulfillment of the Requirements of     */
/*        the Degree of Master of Applied Science        */
/*                                                       */
/*                  McMaster University                  */
/*           Department of Computing and Software        */
/*                 1280, Main Street West                */
/*                  Hamilton, Ontario                    */
/*                    Canada L8S 4K1                     */
/*                                                       */
/*        Copyright (C) 2006  McMaster University        */
/*                                                       */
/*********************************************************/
-->
<!DOCTYPE PMML>

<PMML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://www.dmg.org/PMML-3_1 ../pmml-3-1.xsd"
xmlns="http://www.dmg.org/PMML-3_1"
version="3.1">

<Header copyright="Copyright (c) McMaster University, 2006. All
rights reserved."></Header>

<DataDictionary numberOfFields="3">
    <DataField displayName="TDS" dataType="float" name="TDS"
    isCyclic="0" optype="continuous"> </DataField>
    <DataField displayName="C-BLUE" dataType="string" name="C-BLUE"
    isCyclic="0" optype="categorical">
        <Value displayValue="present" property="valid"
        value="present" />
        <Value displayValue="absent" property="valid"
        value="absent" />
    </DataField>

    <DataField displayName="DIAGNOSIS" dataType="string"
    name="DIAG" isCyclic="0" optype="categorical">
        <Value displayValue="Benign-nevus" property="valid"
        value="Benign-nevus" >
            <Extension extender="CAS" name="Description"
            value="TDS is low and C-Blue is absent, so the
            result of classification is Benign-nevus.">
            </Extension>
        </Value>
```

```
<Value displayValue="Blue-nevus" property="valid"
value="Blue-nevus" >
    <Extension extender="CAS" name="Description"
    value="TDS is low and C-Blue is present, so the
    result of classification is Blue-nevus.">
    </Extension>
</Value>

<Value displayValue="Malignant" property="valid"
value="Malignant" >
    <Extension extender="CAS" name="Description"
    value="TDS is high, so the result of classification
    is Blue-nevus.">
    </Extension>
</Value>
<Value displayValue="Suspicious" property="valid"
value="Suspicious" >
    <Extension extender="CAS" name="Description"
    value="TDS is in an undecidable range, so the
    result of classification is suspicious.">
    </Extension>
</Value>

<Value displayValue="UNKNOWN" property="valid"
value="UNKNOWN" >
    <Extension extender="CAS" name="Description"
    value="The algorithm can not decide.">
```

```
            </Extension>
        </Value>
    </DataField>
</DataDictionary>

<TreeModel modelName="Decision Tree Model"
splitCharacteristic="multiSplit" algorithmName="decisionTree"
functionName="classification">
    <MiningSchema>
        <MiningField name="TDS" usageType="active" />
        <MiningField name="C-BLUE" usageType="active" />
        <MiningField name="DIAG" usageType="predicted" />
    </MiningSchema>

    <Node recordCount="245" score="UNKNOWN">
        <True />
        <ScoreDistribution recordCount="62"
        value="Benign-nevus" />
        <ScoreDistribution recordCount="59"
        value="Blue-nevus" />
        <ScoreDistribution recordCount="62"
        value="Malignant" />
        <ScoreDistribution recordCount="62"
        value="Suspicious" />
        <ScoreDistribution recordCount="0"
        value="UNKNOWN" />
        <Node recordCount="121" score="UNKNOWN">
    <SimplePredicate operator="lessOrEqual"
    value="4.85" field="TDS" />
    <ScoreDistribution recordCount="62"
    value="Benign-nevus" />
    <ScoreDistribution recordCount="59"
    value="Blue-nevus" />
    <Node recordCount="62" score="Benign-nevus">
        <SimplePredicate operator="equal"
        value="absent" field="C-BLUE" />
        <ScoreDistribution recordCount="62"
        value="Benign-nevus" />
    </Node>
    <Node recordCount="59" score="Blue-nevus">
        <SimplePredicate operator="equal"
        value="present" field="C-BLUE" />
        <ScoreDistribution recordCount="59"
        value="Benign-nevus" />
    </Node>
</Node>
<Node recordCount="124" score="UNKNOWN">
    <SimplePredicate operator="greaterThan"
    value="4.85" field="TDS" />
    <ScoreDistribution recordCount="62"
    value="Malignant" />
    <ScoreDistribution recordCount="62"
    value="Suspicious" />
    <Node recordCount="62" score="Malignant">
```

```
                    <SimplePredicate operator="greaterThan"
                    value="5.54" field="TDS" />
                    <ScoreDistribution recordCount="62"
                    value="Malignant" />
                </Node>
                <Node recordCount="62" score="Suspicious">
                    <SimplePredicate operator="lessOrEqual"
                    value="5.54" field="TDS" />
                    <ScoreDistribution recordCount="62"
                    value="Suspicious" />
                </Node>
            </Node>
        </Node>
    </TreeModel>
</PMML>
```

A.6 Sample CDA results document

The following code is a sample CDA results document for the Melanoma skin cancer classifier. The results document is the output of the logic module that receives appropriate data items and applies the decision tree classifier that is encoded as a PMML data mining model.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    This file refers to the input file used to get these results.
-->
<ClinicalDocument
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:hl7-org:v3 ../EMS/EMS_CDA.xsd"
 xmlns:hl7="urn:hl7-org:v3"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="urn:hl7-org:v3">
<!--
    ************
    CDA Header
    ************
-->
    <id extension="6823AC2F-B374-4214-AF1B-407091BBED37"
    root="2.16.840.1.113883.3.933"/>
    <code code="11488-4"
    codeSystem="2.16.840.1.113883.6.1"
    codeSystemName="LOINC" displayName="Consultation"/>
    <title>
        The output document generated by applying the
        "Melanoma skin cancer classifier" on patient data.
    </title>
    <effectiveTime value="20060531121540"/>
    <confidentialityCode code="N"
    codeSystem="2.16.840.1.113883.5.25"/>
<!--
    ************
    Information Recipient
    ************
-->
```

```
    <informationRecipient typeCode="PRCP">
        <intendedRecipient classCode="ASSIGNED">
            <id extension="ksartipi"
            root="DCCD2C68-389B-44c4-AD99-B8FB2DAD1493"/>
            <informationRecipient>
                <name>
                    <prefix>Dr.</prefix>
                    <given>Kamran</given>
                    <given></given>
                    <family>Sartipi</family>
                    <suffix></suffix>
                </name>
            </informationRecipient>
        </intendedRecipient>
    </informationRecipient>
<!--
    ************
    Author
    ************
-->
    <author>
        <time value="20040812120000"/>
        <assignedAuthor>
            <id extension="hhippocrates"
            root="DCCD2C68-389B-44c4-AD99-B8FB2DAD1493"/>
        </assignedAuthor>
    </author>
    <author>
        <time value="20040812120000"/>
        <assignedAuthor>
            <id extension="hhippocrates"
            root="DCCD2C68-389B-44c4-AD99-B8FB2DAD1493"/>
            <assignedAuthoringDevice>
                <softwareName>oXygen XML</softwareName>
            </assignedAuthoringDevice>
        </assignedAuthor>
    </author>
<!--
    ************
    Custodian
    ************
-->
    <custodian typeCode="CST">
        <assignedCustodian>
            <representedCustodianOrganization>
                <id extension="888"
                root="7EEF0BCC-F03E-4742-A736-8BAC57180C5F"/>
                <name>CAS at McMaster University</name>
            </representedCustodianOrganization>
        </assignedCustodian>
    </custodian>
<!--
    ************
    Record Target
    ************
-->
    <recordTarget typeCode="RCT" contextControlCode="OP">
        <patient classCode="PAT">
            <id extension="9999999999"
            root="2BFBA1E9-79C2-4bbb-B589-41B949BD6A3B"
            assigningAuthorityName="PHN"/>
            <patientPatient>
                <name>
                    <prefix>Mr.</prefix>
                    <given>Reza</given>
                    <given></given>
                    <family>Sherafat</family>
                    <suffix></suffix>
                </name>
                <administrativeGenderCode code="M"
                codeSystem="2.16.840.1.113883.5.1"
                codeSystemName="HL7 - Administrative Gender"
                displayName="Male"/>
                <birthTime value="19820622"/>
            </patientPatient>
        </patient>
    </recordTarget>
<!--
    ************
    Related Document
    ************
```

```
-->
    <relatedDocument typeCode="APND">
        <parentDocument>
            <id extension="6823AC2F-B374-4214-AF1B-407091BBED37"
            root="2.16.840.1.113883.3.933"/>
            <code code="34140-4"
            codeSystem="2.16.840.1.113883.6.1"
            codeSystemName="LOINC"
            displayName="Referral"/>
            <text>A previous referral is related to
            this document.</text>
        </parentDocument>
    </relatedDocument>
<!--
    ************
    Structured Body
    ************
-->
    <component>
        <structuredBody>
<!--
    ************
    Purpose
    ************
-->
            <component>
                <section>
```

```
                    <code code="1991"
                    codeSystem="2.16.840.1.113883.6.2"
                    codeSystemName="CAS Lab Observation Table"
                    displayName="Purpose"/>
                    <title>Purpose Section</title>
                    <text>
                        <paragraph>
                            Results of the consultation to
                            the Melanoma skin cancer classifier.
                        </paragraph>
                        <paragraph>
                            Urgency: <content styleCode="bold">
                            Expedite (Call)</content>
                        </paragraph>
                        <paragraph>
                            This document shows whether the
                            patient is diagnosed with Melanoma
                            or not.
                        </paragraph>
                    </text>
                </section>
            </component>

<!--
    ************
    Alerts
    ************
```

```
-->
            <component>
                <section>
                    <code code="1292"
                    codeSystem="2.16.840.1.113883.6.2"
                    codeSystemName="CAS Lab Observation Table"
                    displayName="Alerts"/>
                    <title>Alerts</title>
                    <text>
                        <table border="1">
                            <tbody>
                                <tr>
                                    <th>Alert</th>
                                    <th>Comments</th>
                                    <th>Date</th>
                                </tr>
                                <tr>
                                    <td>The result is malignant
                                    Melanoma.</td>
                                    <td>TDS is high, so the
                                    result of classification
                                    is Blue-nevus.</td>
                                    <td>May 31, 2006</td>
                                </tr>
                            </tbody>
                        </table>
                    </text>
                    <entry typeCode="COMP">
                        <observation classCode="ALRT"
                        moodCode="INT">
                            <code code="2345"
                            displayName="Malignant Melanoma"
                            codeSystemName="CAS Lab Observation Table"
                            codeSystem="2.16.840.1.113883.6.2">
                            </code>
                        </observation>
                    </entry>
                </section>
            </component>
        </structuredBody>
    </component>
</ClinicalDocument>
```

A.7 XPath expressions to access data items from the results CDA instance document

Table A.4 lists the XPath expressions used to access the data items from the results CDA document of the Melanoma skin cancer classifier.

Data item	XPath
Skin cancer's mark diversity	`/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/ hl7:component/hl7:section/ hl7:entry[@typeCode='COMP']/ hl7:observation[@moodCode='EVN' and @classCode='OBS']/hl7:code [@code='1234-4' and @codeSystem='2.16.840.1.113883.6.2']/../ hl7:entryRelationship[@typeCode='COMP']/ hl7:observation[(@classCode='OBS') and (@moodCode='EVN')]/ hl7:code/attribute::code`
C-Blue	`/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/ hl7:component/hl7:section/ hl7:entry[@typeCode='COMP']/ hl7:observation[@moodCode='EVN' and @classCode='OBS']/ hl7:code[@code='1234-5' and @codeSystem='2.16.840.1.113883.6.2']/../ hl7:entryRelationship[@typeCode='COMP']/ hl7:observation[(@classCode='OBS') and (@moodCode='EVN')]/hl7:code/attribute::code`

Table A.3: XPath expressions used to access the data items for the Melanoma skin cancer classifier.

Item	XPath
Description of the result	`/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/hl7:component/ hl7:section/code[@code='1292' and @codeSystem='2.16.840.1.113883.6.2']/ ../hl7:text`
Coded value of the result	`/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/hl7:component/ hl7:section/code[@code='1292' and @codeSystem='2.16.840.1.113883.6.2']/ ../hl7:entry[@typeCode='COMP']/ observation[@classCode='ALRT' and @moodCode='INT']/code/attribute::code`
CodeSystem code of the result	`/hl7:ClinicalDocument/hl7:component/ hl7:structuredBody/hl7:component/ hl7:section/code[@code='1292' and @codeSystem='2.16.840.1.113883.6.2']/ ../hl7:entry[@typeCode='COMP']/ observation[@classCode='ALRT' and @moodCode='INT']/code/attribute::codeSystem`

Table A.4: XPath expressions used to access the results of classification from the results CDA documents.

Bibliography

[1] International Classification of Diseases (ICD). URL = http://www3.who.int/icd/vol1htm2003/fr-icd.htm. [Online; accessed 1-August-2006].

[2] Health Level-7. URL = http://www.hl7.org. [Online; accessed 1-August-2006].

[3] *Protégé* ontology editor tool. URL = http://protege.stanford.edu/. [Online; accessed 1-August-2006].

[4] Data Management Group (DMG) website. URL = http://www.dmg.org/. [Online; accessed 1-August-2006].

[5] Logical Observation Identifiers Names and Codes (LOINC). URL = http://www.regenstrief.org/loinc/. [Online; accessed 1-August-2006].

[6] The ASGAARD project page. URL = http://www.asgaard.tuwien.ac.at/. [Online; accessed 1-August-2006].

[7] Medical Subject Headings (MeSH). URL = http://www.nlm.nih.gov/mesh/. [Online; accessed 1-August-2006].

[8] Systematized Nomenclature of Medicine Clinical Terminology (SNOMED CT). URL = http://www.snomed.org/snomedct/index.html. [Online; accessed 1-August-2006].

[9] The Arden Syntax for Medical Logic Systems, URL = http://cslxinfmtcs.csmc.edu/hl7/arden/. [Online; accessed 1-August-2006].

[10] Rules for Melanoma skin cancer diagnosis, URL = http://www.phys.uni.torun.pl/publications/kmk/. [Online; accessed 1-August-2006].

[11] Unified Medical Language System (UMLS). URL = http://www.nlm.nih.gov/research/umls/. [Online; accessed 1-August-2006].

[12] Computerization Of Medical Practices for the Enhancement of Therapeutic Effectiveness (COMPETE). URL = http://www.compete-study.com. [Online; accessed 1-August-2006].

[13] Prudsys AG. XELOPES library documentation - version 1.3.1. URL = http://www.prudsys.com/Service/Downloads/bin/1133983554/Xelopes1.3.1_Intro.pdf. [Online; accessed 1-August-2006].

[14] American Hospital Association (AHA). *Hospital Statistics.* Health Forum Publishing Company, 1999.

[15] V. Anand, PG. Biondich, G. Liu, M. Rosenman, and SM. Downs. Child Health Improvement through Computer Automation: The CHICA system. *MedInfo*, 2004;2004:187-91.

[16] Australia's National Electronic Decision Support Taskforce. Electronic decision support for Australia's health sector. URL = http://www.ahic.org.au/downloads/nedsrept.pdf. [Online; accessed 1-August-2006], January 2003.

[17] Guideline Interchange Format (GLIF)3.5 - technical specification. URL = http://smi-web.stanford.edu/projects/intermed-web/guidelines/GLIF_TECH_SPEC_May_4_2004.pdf. [Online; accessed 1-August-2006], May 2004.

[18] Pavel Berkhin. Survey of clustering data mining techniques. URL = http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf. [Online; accessed 1-August-2006].

[19] Canadian Institute for Health Informatics (CIHI). Health expenditure in Canada. URL = http://www.cihi.ca/cihiweb/dispPage.jsp?cw_page=media_07dec2005_e. [Online; accessed 1-August-2006], December 2005.

[20] Leonid Churilov, Adil M. Bagirov, D. Schwartz, Kate A. Smith, and M. Dally. Improving risk grouping rules for prostate cancer patients with optimization. In *Hawaii International Conference on System Sciences (HICSS)*, 2004.

[21] Data Management Group (DMG). Predictive Model Markup Language (PMML) version 3.0 specification. URL = http://www.dmg.org/pmml-v3-0.html.

[22] Brendan C. Delaney, David A. Fitzmaurice, Amjid Riaz, and F.D. Hobbs, Richard. Can computerised decision support systems deliver improved quality in primary care? *BMJ*, 319(7220):1281, 1999.

[23] EGADSS.org. Evidence-based Guidelines And Decision Support System (EGADSS) project page. URL = http://egads.org. [Online; accessed 1-August-2006].

[24] Electronic-Medical Summary (e-MS). The e-MS project page. URL = http://www.e-ms.ca/. [Online; accessed 1-August-2006].

[25] Electronic-Medical Summary (e-MS). The e-MS standard specification. URL = http://www.e-ms.ca/. [Online; accessed 1-August-2006].

[26] Cynthia M. Farquhar, Emma W. Kofa, and Jean R. Slutsky. Clinicians' attitudes to clinical practice guidelines: a systematic review. *Medical Journal of Australia*, 177(9):502–506, 2002.

[27] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54, 1996.

[28] I. Haschler, S. Skonetzki, HJ. Gausepohl, and O. Linderkamp and. T Wetter. Evolution of the HELEN representation for managing clinical practice guidelines. *Submitted to Methods of Information in Medicine*, May 2005.

[29] I. Haschler, S. Skonetzki, HJ. Gausepohl, and O. Linderkamp and. T Wetter. Evolution of the HELEN representation for managing clinical practice guidelines. *Submitted to Computer Methods & Programs in Biomedicine*, September 2005.

[30] Health Level 7. The Clinical Document Architecture (CDA) standard specification. URL = `http://www.hl7.org`. [Online; accessed 1-August-2006].

[31] Health Level 7. Health Level-7 Reference Information Model (HL-7 RIM). URL = http://www.hl7.org.

[32] Dereck L. Hunt, R. Brian Haynes, Steven E. Hanna, and Kristina Smith. Effects of computer-based Clinical Decision Support Systems on physician performance and patient outcomes. *Journal of the American Medical Association*, 280:13391346, 1998.

[33] IBM. The Healthcare Collaborative Network (HCN). URL = `http://www-03.ibm.com/industries/healthcare/doc/content/landing/972420105.html`. [Online; accessed 1-August-2006].

[34] Walker H. Land Jr, Timothy Masters, Joseph Y. Lo, Daniel W. McKee, and Frances R. Anderson. New results in breast cancer classification obtained from an evolutionary computation/adaptive boosting hybrid using mammogram and history data. In *2001 IEEE Mountain Workshop on Soft Computing in Industrial Applications*, pages 47–52, 2001.

[35] Parameshvyas Laxminarayan, Carolina Ruiz, Sergio A. Alvarez, and Majaz Moonis. Mining associations over human sleep time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS 2005)*, pages 323–328, 2005.

[36] Jiuyong Li, Ada Wai-Chee Fu, Hongxing He, Jie Chen, Huidong Jin, Damien McAullay, Graham Williams, Ross Sparks, and Chris Kelman. Mining risk patterns in medical data. In Robert Grossman, Roberto Bayardo, and Kristin P. Bennett, editors, *KDD*, pages 770–775. ACM, 2005.

[37] Object Management Group (OMG). Healthcare Data Interpretation Facility (HDIF). URL = `http://www.omg.org/docs/corbamed/98-03-07.pdf`. [Online; accessed 1-August-2006], August 1998.

[38] Ontario Ministry of Finance. The right choices: Investing in health care. URL = `http://www.fin.gov.on.ca/english/budget/bud03/budhi1.html`. [Online; accessed 1-August-2006], March 2003.

[39] Carlos Ordonez, Cesar A. Santana, and Levien de Braal. Discovering interesting association rules in medical data. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 78–85, 2000.

[40] PRESGUID project page. URL = `http://cybertim.timone.univ-mrs.fr/CybErtim/LERTIM/Recherche/PRESGUID/contenu.htm`. [Online; accessed 1-August-2006].

[41] Eric J. Thomas, David M. Studdert, Joseph P. Newhouse, Brett I. W. Zbar, K. Mason Howard, Elliott Williams, and T. Brennan. Costs of medical injuries in Utah and Colorado. 36:255–264, 1999.

[42] Antonia Vlahou, John O. Schorge, Betsy W. Gregory, and Robert L. Coleman. Diagnosis of ovarian cancer using decision tree classification of mass spectral data. *Journal of Biomedicine and Biotechnology*, 5:308–314, 2003.

[43] Dongwen Wang, Mor Peleg, Samson W. Tu, Aziz A. Boxwala, Omolola Ogunyemi, Qing Zeng, Robert A. Greenes, Vimla L. Patel, and Edward H. Shortliffea. Design

and implementation of the GLIF3 guideline execution engine. *Journal of biomedical informatics*, 2004 Oct;37(5):305-18.

[44] Prof. Dr. Th. Wetter. HELEN. URL = `http://www.klinikum.uni-heidelberg.de/Englische-Version-Promotionsarbeiten.8915.0.html`. [Online; accessed 1-August-2006], 2005.

[45] Wikipedia. Health Insurance Portability and Accountability Act (HIPAA)—wikipedia, the free encyclopedia., 2006. URL = `http://en.wikipedia.org/w/index.php?title=Health_Insurance_Portability_and_Accountability_Act&oldid=66219756`. [Online; accessed 1-August-2006].

[46] Andrew M. Wilson, Lehana Thabane, and Anne Holbrook. Application of data mining techniques in pharmacovigilance. *British Journal of Clinical Pharmacology*, 57(2):127–34, Feb 2004.

[47] Osmar R. Zaïane, Maria-Luiza Antonie, and Alexandru Coman. Mammography classification by an association rule-based classifier. In *MDM/KDD*, pages 62–69, 2002.