NOTE TO USERS

Duplicate page number(s); text follows.
The manuscript was scanned as received.

83

This reproduction is the best copy available.

UMI[®]

A Health Care Framework for Medical Data Interoperability & Clinical Decision Support System Integration Using XML

Christina Anne Catley

A thesis submitted to

The Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

In Electrical Engineering

Department of Systems and Computer Engineering

Carleton University

1125 Colonel By Drive

Ottawa, Ontario, K1S 5B6

Canada

© July, 2002

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-79694-9

Canada

The undersigned recommend to the
Faculty of Graduate Studies and Research

Acceptance of the thesis:

**A Health Care Framework for Medical Data
Interoperability & Clinical Decision Support System
Integration Using XML**

submitted by

Christina Anne Catley

in partial fulfilment of the requirements

for the degree of

M.A.Sc. in Electrical Engineering

Professor M. Frize, Thesis Supervisor

Chair, Department of Systems and Computer Engineering

Carleton University

© July, 2002

The author wishes to thank her parents:

To Bruce for always being there to support me in achieving my goals,
and

to Gail for instilling the idea that her daughters can do anything.

To Andrea, sister, best friend and long-distance phone confidante.

And of course, to my husband Todd –
the endlessly patient and wonderful man who offered constant
love and encouragement.

Abstract

This thesis presents a framework for promoting medical data interoperability and for integrating diverse clinical decision support systems using the eXtended Markup Language (XML). The current lack of standardized frameworks for managing and sharing medical data in enterprise applications has resulted in numerous ‘isolated silos’ of information. The motivation for this work is that these isolated silos of non-standardized data severely hinder the widespread adoption and integration of clinical decision support systems, such as those developed within the Medical IDEAs Research Group. As a growing number of studies suggest that clinical decision support systems have a positive impact on reducing medical errors, the potential significance of integrated clinical decision support systems becomes evident. In addition to modeling the framework’s architectural and technological issues, the framework is described at the contextual and conceptual level in order to highlight its broader objectives and to hopefully obtain consensus and corroboration on these objectives in the medical informatics community.

Acknowledgements

I would like to begin by thanking my thesis supervisor, Dr. Monique Frize, for her ongoing guidance as well as the energy and enthusiasm she directed into this work. On a general level, I am grateful to Dr. Frize for introducing me to the field of medical informatics, and on a personal note I have especially appreciated her ability to build confidence in her students. Thanks also to Dr. Frize for financial support of this work provided through her NSERC grant #202972-01.

Thank you as well to members of the Medical IDEAS Research Group, particularly to Colleen Ennett for offering encouragement and answering all my ‘thesis-related’ questions over the past two years, and to Maimouna Bagna for volunteering to aid in standardizing MIRG’s medical databases.

Finally, very special thanks are reserved for Bruce Catley, for reading and editing thesis drafts- your time, effort and input were tremendously appreciated (especially when I am sure you would have rather been sailing!).

Contents

Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	x
List of Tables	xii
Glossary of Terms and Acronyms	xiii
Chapter 1 Introduction	1
1.1 MOTIVATION AND SIGNIFICANCE OF RESEARCH.....	2
1.1.1 A HEALTH CARE PERSPECTIVE	3
1.1.2 MAJOR TRENDS IN HEALTH CARE INFORMATION TECHNOLOGY	3
1.2 THESIS OBJECTIVES.....	5
1.3 THESIS CONTRIBUTIONS.....	7
1.4 THESIS OUTLINE	8
Chapter 2 Overview of Relevant Medical and Technological Concepts	10
2.1 MIRG'S MEDICAL SOFTWARE AND APPLICATION ENVIRONMENT.....	10
2.1.1 ARTIFICIAL NEURAL NETWORKS	11
2.1.2 CASE-BASED REASONING.....	14
2.1.3 CHEO'S NICU REAL-TIME DATA REPOSITORY	15
2.1.4 CHALLENGES FACED IN THE CURRENT ENVIRONMENT	16
2.2 EXTENDED MARKUP LANGUAGE (XML)	17
2.2.1 OVERVIEW OF THE EXTENDED MARKUP LANGUAGE	18
2.2.2 XML SCHEMA LANGUAGE.....	21
2.2.3 EXTENDED STYLESHEET LANGUAGE TRANSFORMS (XSLT).....	22
2.2.4 XML PARSERS.....	23
2.2.5 XML AND DATABASES.....	24
2.2.6 IN WHICH SITUATIONS IS XML AN APPLICABLE TECHNOLOGY?	25
Chapter 3 Literature Review	27
3.1 ARCHITECTURES FACILITATING INTEROPERABILITY OF DISTRIBUTED MEDICAL DATA AND MEDICAL APPLICATIONS.....	28
3.1.1 FUNCTIONAL GOAL #1: INTEGRATING DISTRIBUTED INFORMATION SOURCES.....	29

3.1.1.1	The SynEx Project*	30
3.1.1.2	The EDRN Project*	31
3.1.1.3	Internet-Based 3D Radiotherapy Planning and Information System (IRIS)*	34
3.1.2	FUNCTIONAL GOAL #2: STANDARDIZED INFORMATION EXCHANGE WITHIN AND BETWEEN HOSPITAL INFORMATION SYSTEMS	36
3.1.2.1	The Hospital Information System DbMap Project (DbMap)*	36
3.1.2.2	Universidad Politecnica's Medical Database Migration (UPMDM) Project	38
3.1.2.3	Integrated Medical Information Systems (IntMedInfS) Project	40
3.1.2.4	Geneva Hospital's HIS/Clinical Information Project (GenHos-CIS)	42
3.1.3	FUNCTIONAL GOAL #3: INTERFACING BETWEEN EPR(S) AND A CDSS	44
3.1.3.1	Medical College of Wisconsin's Integration of Bayesian Networks and Structured Reporting (IntBNWStrucRep) Project	44
3.1.3.2	University of Malaysia's Case-Based Reasoning Interaction with Electronic Patient Record Repositories (CBR-EPR) Project	45
3.1.4	FUNCTIONAL GOAL #4: INTEGRATION OF CLINICAL DECISION SUPPORT IN A HIS	47
3.1.4.1	Integration of CDSS into two Hospital Information Systems (CDS-HIS)	47
3.2	CURRENT MEDICAL STANDARDIZATION EFFORTS	50
3.2.1	HEALTH LEVEL 7 (HL7)	52
3.2.2	UNIFIED MEDICAL LANGUAGE SYSTEM (UMLS)	53
3.2.3	MEDICAL XML VOCABULARIES	54
3.3	OPEN SOURCE DEVELOPMENT OF MEDICAL SYSTEMS	56
3.3.1	WHAT CONSTITUTES AN OPEN SOURCE SYSTEM?	57
3.3.2	OASIS AND 'XML.ORG'	57
3.3.3	THE OPEN SOURCE SOFTWARE HEALTH CARE ALLIANCE (OSHCA)	58
3.3.4	INTEROPERABILITY OF OPEN SOURCE SYSTEMS	60
Chapter 4 Designing a Framework for Integrated Clinical Decision Support and Medical Data Interoperability		62
4.1	PROBLEM STATEMENT	63
4.1.1	CONTEXTUAL SCOPE OF THE CDSS FRAMEWORK	65
4.1.2	ARCHITECTURAL GOALS AND PRINCIPLES	67
4.1.3	BUSINESS MODELING OF THE MEDICAL DOMAIN	71
4.2	CRITICAL ANALYSIS OF MEDICAL INTEGRATION INITIATIVES	75

4.2.1	COMPARISON OF FUNCTIONAL GOALS	76
1.1.2	COMPARISON OF TECHNOLOGICAL ENABLERS	83
1.3	PROBLEM SIGNIFICANCE.....	84
1.3.1	SIGNIFICANCE FROM A HEALTH CARE PERSPECTIVE	84
1.3.1.1	The ‘Silos of Information’ Paradigm	84
1.3.2	SIGNIFICANCE FROM THE MEDICAL IDEAS RESEARCH GROUP’S PERSPECTIVE	87
Chapter 5 Framework Architecture and Technology Validation		90
5.1	REQUIREMENTS MODEL.....	90
5.1.1	USE CASE MODEL: ACTOR IDENTIFICATION	91
5.1.2	USE CASE MODEL: USE CASE IDENTIFICATION	92
5.1.2.1	MIRG CDSS Processing Package: Use Case Diagram and Descriptions.....	94
5.1.2.1.1	Invoke CDSS Processing	95
5.1.2.1.2	CDSS Semantic Interoperability	97
5.1.2.1.3	Hybrid CDSS Processing	98
5.1.2.1.4	User Registration	99
5.1.2.1.5	Authenticate User and Establish Access Parameters	100
5.1.2.1.6	Validate Data	100
5.1.2.1.7	Transform Medical Data	101
5.1.2.1.8	Process Medical Data by CDSS	102
5.1.2.1.9	Generate Interoperable and Standardized Results	103
5.1.2.1.10	Generate Integrated Response	104
5.1.2.1.11	Generate CDSS-Driven Alerts	104
5.1.2.2	MIRG Interoperability Support Package: Use Case Diagram and Descriptions.....	105
5.1.2.2.1	Update MIRG Medical Schema Repository	106
5.1.2.2.2	Update MIRG HIS Schemas	106
5.1.2.2.3	Publish Schemas in Registry/Repository	107
5.1.2.2.4	System Maintenance	107
5.1.2.2.5	Secure Login	108
5.1.2.2.6	Access Standardized Metadata via MIRG Repository	108
5.1.2.2.7	Standardize Medical Data	109
5.1.2.2.8	Manual Standardization	110
5.1.2.3	NICU CDSS Support Package: Use Case Diagram and Descriptions	110
5.1.2.3.1	Prepare NICU Repository Data for CDSS Processing	111
5.1.2.3.2	Device Scheduling (Real-time/interrupt driven and polled)	111
5.1.2.3.3	Real-time or Polled update of NICU Medical Data Repository	112
5.1.3	NON-FUNCTIONAL SYSTEM REQUIREMENTS.....	112
5.1.3.1	System Distribution (Network Topology)	113
5.1.3.2	Usability	113
5.1.3.3	Reliability	113

5.1.3.4	Performance	114
5.1.3.5	Design Constraints	114
5.1.3.6	Security	115
5.1.3.7	Interfaces	115
5.2	ANALYSIS MODEL	116
5.2.1	CLASS DESIGN AND PACKAGING STRATEGY.....	116
5.2.1.1	Boundary Classes	116
5.2.1.2	Entity Classes	118
5.2.1.3	Control Classes.....	120
5.2.1.3.1	MIRG Server	121
5.2.1.3.2	NICU Client	124
5.2.1.3.3	UMLS Server	126
5.3	DESIGN MODEL	126
5.3.1	DEPLOYMENT MODEL	127
5.3.1.1	Justification for Choice of Distribution Technology	129
5.3.1.2	XML Messaging-based Distributed Computing	131
5.3.1.3	Data Security	132
5.3.2	ARCHITECTURAL CONSIDERATIONS (DISTRIBUTION PATTERNS)	135
5.3.2.1	Client-side Considerations	137
5.3.2.2	Data Distribution	138
5.3.2.3	Server-side CDSS Processing: Managing High-Volume Throughput	140
5.3.3	DESIGN ARTEFACTS.....	142
5.3.3.1	MIRG Application Server Subsystem.....	145
5.3.3.2	NICU Application Client Subsystem	150
5.3.3.3	Business Domain Layer	151
5.3.3.4	Middleware	153
5.4	TECHNOLOGY VALIDATION	154
5.4.1	STANDARDIZING MEDICAL DATABASES BASED ON THE UMLS.....	158
5.4.2	XML ENABLING THE DATA: XML DOCUMENTS, SCHEMAS AND DTDs	160
5.4.2.1	Parsing and Validating XML Data.....	161
5.4.2.2	Schema Repositories	162
5.4.3	STORING DATA RECEIVED FROM XML-BASED APPLICATIONS	163
5.4.4	INTEGRATION WITH MIRG'S CLINICAL DECISION SUPPORT SYSTEMS.....	165
5.4.5	CONCLUSIONS BASED ON TECHNOLOGICAL VALIDATION.....	166

Chapter 6 Conclusion	168
6.1 CONCLUDING REMARKS.....	168
6.2 THESIS CONTRIBUTIONS.....	169
6.2.1 CONTRIBUTIONS TO KNOWLEDGE	169
6.2.2 PRACTICAL CONTRIBUTIONS	170
6.3 FUTURE WORK.....	171
Appendix A Unified Modeling Language Overview	175
Appendix B Zachman Framework Matrix.....	179
References	180

List of Figures

Figure 2-A	Data Flow in Ann Pre-processing	13
Figure 3-A	SynEx Architecture [Xu <i>et al.</i> 2001]	31
Figure 3-B	SynEx Use Case Diagram	31
Figure 3-C	EDRN Architecture [Crichton <i>et al.</i> 2001]	33
Figure 3-D	EDRN Use Case Diagram	33
Figure 3-E	IRIS Architecture [Luttgau & Bendl 2001]	35
Figure 3-F	IRIS Use Case Diagram	35
Figure 3-G	DBMAP Architecture [Karadimas <i>et al.</i> 2000]	37
Figure 3-H	DbMAP Use Case Diagram	38
Figure 3-I	UPMDM Architecture [Pons <i>et al.</i> 1999]	40
Figure 3-J	UPMDM Use Case Diagram	40
Figure 3-K	IntMedInfS Use Case Diagram	42
Figure 3-L	GenHos-CIS Architecture [Geissbuhler <i>et al.</i> 2001]	43
Figure 3-M	GenHos-CIS Use Case Diagram	43
Figure 3-N	IntBNWStrucRep Use Case Diagram	45
Figure 3-O	CBR-EPR Architecture [Manickam & Abidi 2001]	46
Figure 3-P	CBR-EPR Use Case Diagram	47
Figure 3-Q	CDS-HIS Architecture	49
Figure 3-R	CDS-HIS Use Case Diagram	50
Figure 4-A	MIRG Network Connectivity Goals	67
Figure 4-B	Medical Business Domain: Interoperability Support Use Case Package	73
Figure 4-C	Medical Business Domain: CDSS-Interactions	74
Figure 4-D	Broad Web-based Functionality	75
Figure 5-A	MIRG CDSS Processing Use Case Diagram	94
Figure 5-B	Activity Diagram for Invoke CDSS Processing	97
Figure 5-C	MIRG Interoperability Support	105

Figure 5-D NICU CDSS Support	110
Figure 5-E Boundary Classes	118
Figure 5-F Persistence Classes (Persistence Package)	120
Figure 5-G Collaboration Diagram (Control Classes).....	123
Figure 5-H Registration and Resource Management Class Diagram	124
Figure 5-I NICU Client Class Diagram	125
Figure 5-J UMLS Classes.....	126
Figure 5-K Deployment Model	128
Figure 5-L System Layering and Package Overview	144
Figure 5-M MIRG Application Server Class Diagram	148
Figure 5-N MIRG Application Server Sequence Diagram	149
Figure 5-O NICU Application Client Classes.....	151
Figure 5-P Business Entities	153
Figure 5-Q Implementation Overview	157
Figure A-A Use Case Relationships	175
Figure A-B Activity Diagram	176
Figure A-C Class Diagram	177
Figure A-D Class Stereotypes	177
Figure A-E Sequence Diagram.....	178
Figure A-F Package Diagram	178

List of Tables

Table 2-A	Communication Levels at which XML can Support Interoperability	26
Table 4-A	Zachman's Framework: Enterprise Scope	66
Table 4-B	Architectural Goals for MIRG's Integrated CDSS Framework.....	68
Table 4-C	Architectural Principles for Medical Information Systems.....	70
Table 4-D	IEEE Interoperability Definitions	77
Table 4-E	Functional Comparison based on Medical Business Model's Interoperability Support Package (section 4.1.3)	79
Table 4-F	Functional Comparison Based on Medical Business Model's CDSS Interactions (section 4.1.3)	81
Table 4-G	Functional Comparison Based on Medical Business Model's Web-based Functionality (section 4.1.3)	82
Table 4-H	Technological Comparisons	83
Table 5-A	Preliminary Actor Identification	91
Table 5-B	Actor Roles.....	92
Table 5-C	UMLS-Standardized Medical Database and Schema Information	159

Glossary of Terms and Acronyms

Actor: In UML terminology an actor is an outside user who interacts with the system.

ANN: Artificial Neural Network

Architecture: The set of significant decisions about the organisation of a software system, the selection of the structural elements and their interfaces by which the system is composed together with their behaviour [Jacobson *et al.* 1999].

ASE: Average Squared Error

BSML: Bioinformatic Sequence Markup Language

CBR: Case-Based Reasoner

CCR: Correct Classification Rate

CDSS: Clinical Decision Support System

CHEO: Children's Hospital of Eastern Ontario

CIHI: Canadian Institute for Health Information

Class: A UML term to describe objects that share the same operations, attributes and relationships. Classes can be thought of as the blueprint for the object.

Client: A user requesting the services provided by the server's interface.

CML: Chemical Markup Language

CORBA (Common Object Request Broker Architecture): An open systems standard developed by the Object Management Group for middleware technology; the standard allows communication between distributed objects on heterogeneous platforms [Gomaa 2000].

DBMS: Database Management System

DICOM: Digital Imaging and COmmunication in Medicine

Distributed client-server architecture: An architecture where the applications, business and data services reside on different nodes.

DOM: Document Object Model

DSSSL: Document Style Semantics and Specification Language

DTD (Document Type Definition): As defined in XML 1.0, a DTD is the formal definition of the elements, structures, and rules for enabling platform independent data access via XML or for marking up a given type of SGML document [W3C XML 2000].

EDI: Electronic Data Interchange

Enterprise: An undertaking or venture in the real world.

EPR: Electronic Patient Record

Framework: Defines concepts, organisational structures, architectural components and ideas that are considered most relevant in achieving a vision.

Gateway: A system that is responsible for transforming the conventions used within one system to the conventions used in another.

HIS: Hospital Information System

HIMSS: Health care Information and Management Systems Society

HL7: Health Level 7

HTML (Hypertext Markup Language): The language used in creating a web page. In an HTML document tags, which specify how information is displayed, enclose content.

HTTP (HyperText Transfer Protocol): The standard web transfer protocol.

Integration: Implies both semantic and technical interoperability, as well as a logical integration flow between interoperable modules.

Interoperability: The ability of two or more systems or components to exchange information and to use the information that has been exchanged.

ISO (The International Organization for Standardization): A body dedicated to the production of standards through international agreement.

JAXP: Sun's Java API for XML Parsing

JDBC: Java DataBase Connectivity

Mapping: A set of objects having a defined correspondence with the members of another set.

Medical Informatics: Medical Informatics combines the disciplines of medicine and information technology to support the health care delivery process and improve the quality of care delivered, through initiatives such as CDSSs.

Metadata: Data about data.

Middleware: A layer of software that sits above the operating system to provide a uniform platform above which distributed applications can run [Gomaa 2000].

MIRG: Medical Intelligent DEcision Aids Research Group

MML: Medical Markup Language

MTBF: Mean time between failures

NICU: Neonatal Intensive Care Unit

OASIS: Organization for the Advancement of Structured Information Standards

Object: A concrete realisation of a class; an object is often referred to as an instantiation of a class.

Ontology: Defines vocabularies of representation terms, this can refer to the ontological commitment to using shared vocabularies.

OSHCA: Open Source Software Health Care Alliance

OSI: Open Systems Interconnection

PACS: Picture Archiving and Communication Systems

Parser: A parser is a specialized software program that recognizes XML content and can extract markup in a document.

PKI: Public Key Infrastructure

Portal: A site providing commonly used web services for a particular domain. In the context of this thesis, XML portals are particularly significant.

RIS: Radiology Information Systems

Real-time: A problem, system or application that is concurrent in nature and has timing constraints whereby incoming events must be processed in a given timeframe.

Registry: A facility that stores relevant descriptive information about registered objects, where a registered object is something important that an author or producer wants to have visible to the world so that it can be discovered and used by a customer or client. A registry can operate independently or it can be paired with a repository that stores the registered objects [Gallagher & Carnahan 2000].

Repository: A repository is simply a storage facility for registered objects, with an access method allowing one to retrieve individual objects by an object reference [Gallagher & Carnahan 2000].

RPC (Remote Procedure Call): A procedure call in which the calling procedure and the called procedure do not share an address space and may be in physically different locations.

RUP: Rational Unified Process

SAX: Simple API for XML

Schema: A schema defines the elements that can appear within a document and the attributes that can be associated with an element, as well as defining the document's structure.

SSL: Secure Sockets Layer

Server: A role filled by a processing element when it provides services to another processor.

SGML: Standard Generalized Markup Language

SMP (Symmetric Multiprocessing): A computer architecture that improves system performance by making multiple CPUs available to complete individual processes concurrently.

SOAP: Simple Object Access Protocol

SQL: Structured Query Language

Three-tier (n-tier) architecture: A special case of a client-server architecture in which system functionality is divided into three (or 'n') components: application, business and data services.

UML (Unified Modeling Language): A language for visualising, specifying, constructing and documenting the artefacts of a software intensive system. [Jacobson *et al.* 1999]

UMLS: Unified Medical Language System

Use Case: A UML term describing interactions between actors and the system; the complete set of use cases describes the system functionality.

Validating XML Parser: A parser that reads a DTD or schema and checks and reports on markup errors is a validating XML parser. An XML document that has no markup errors is a valid XML document.

W3C: World Wide Web Consortium

Web Delivery Pattern: A web-based distributed object system; the web browser is a delivery and container mechanism for a distributed object system.

Web Services: An interface that describes a collection of operations that are network accessible through standardized XML messages.

Well-formed XML document: As defined in XML 1.0, a well-formed XML document is one with balanced starting and ending tags.

XML4J: IBM's XML Parser for Java

XML (eXtensible Markup Language): As defined in XML 1.0, XML is a standard from the W3C that provides tagging of information content within documents. XML offers a means for representing content in a format that is both human and machine readable.

XML Namespaces: XML namespaces provide a simple method for qualifying element and attribute names used in XML documents. This is accomplished by associating a particular tag set by associating a prefix with a Universal Resource Identifier (URI) reference.

XSLT (extended Stylesheet Language Transformations): Describes a vocabulary recognized by an XSLT processor to transform information from the format specified in the source file into a different format. Simply put, XSLT takes XML content and transforms it into any other format for which an applicable transform has been created.

“No industry will be more affected by technology than health care”

Don Tapscott, *The Digital Economy*

Chapter 1 Introduction

Medical software development relies heavily on the use of medical databases. Most of these databases are institution specific, and as such have heterogeneous (non-standard) data models and encodings. The majority of existing medical software handles the problem of non-standardized databases by manipulating the data into a ‘unique’ form that is ‘understood’ by the program in question. This practice decreases the reusability of the software, severely limits the potential for information sharing, and increases the probability of programmer error.

The Internet has the potential to facilitate the sharing of medical databases, and is underlying an increasing number of medical record and clinical decision support systems. However, relatively few medical systems adequately address application connectivity and data interoperability issues, and thus fail to realise this potential. The result is a plethora of proprietary medical software and heterogeneous databases, operating for the most part in isolation. As stated by Karadimas: “*Applications that can address multiple heterogeneous (non-standardized) databases with uniform access are difficult to write and maintain*” [Karadimas *et al.* 2000].

Medical informatics experts agree that an “*integrated vision*” [Flesca *et al.* 2000] is needed in order to facilitate the shared usage of the large amount of electronic data available to researchers. McDonald expressed this idea eloquently: “*The sources of patient data that do exist... reside on many isolated islands*” [McDonald 1997]. These ‘isolated islands’ are frequently described in the literature as ‘isolated silos of

information'; information isolation is a particular problem in health care where data confidentiality is imperative and standardization efforts are still in their infancy.

Intensive standardization efforts are required before medical databases can become the shared commodity envisioned by scientists and health care professionals. An ideal medical database would then have broad applicability and would adhere to standardized, global processes that are universally acceptable.

In this thesis, standardized medical databases will form the basis for a proposed Clinical Decision Support System Integration Framework, offering distributed access to standardized medical information and inter-operable medical applications.

1.1 Motivation and Significance of Research

The Medical IDEAS (Intelligent DEcision AidS) Research Group (MIRG) is a multidisciplinary team composed of engineers, physicians, statisticians and students; the team's primary objective is to develop artificial intelligence based clinical decision support system (CDSS) software such as Case-Based Reasoning Systems (CBRs) and Artificial Neural Networks (ANNs) to improve patient care in the hospital environment.

As MIRG software begins to be used in sophisticated distributed and possibly web-based applications and interfaced with real-time databases such as the Children's Hospital of Eastern Ontario's (CHEO's) Neonatal Intensive Care Unit (NICU) real-time data repository, the need for extending the applicability of medical data through information sharing is greatly increased. In effect, the data received from the real-time data repository must support a broad range of applications, since ideally the data should be able to be processed through ANN, CBR or other clinical decision support system software based on the physicians' preferences.

1.1.1 A Health Care Perspective

Before discussing the challenges facing medical data interoperability and application integration of clinical decision support systems, it is insightful to justify the existence of clinical decision support systems in the hospital environment. At present there are two driving forces in the health care sector that are inexorably linked: to improve patient care and to reduce expenditures in the face of rising cost pressures. Clinical decision support systems will play a role in helping health care workers jointly achieve these two potentially contradictory goals.

An interesting example can be found in the NICU, where physicians are often forced to make difficult decisions regarding the potential outcomes of very premature infants based on inconclusive statistical evidence. As Dr. Walker of CHEO explains: in the current health care setting the best a physician can provide in terms of concrete evidence regarding the outcome of a premature infant of less than 24 weeks gestation is to inform parents that their child has a 50% chance of being mentally and/or physically handicapped [Walker 2001]. Physicians do not currently have any means to give specific information about an individual infant. A real-time CDSS that integrated ANN and CBR technology would allow physicians to provide this information based on making comparisons with other infants with similar conditions using the CBR, and by predicting the probability of different outcomes with the ANN.

1.1.2 Major Trends in Health Care Information Technology

MIRG software is ultimately designed for use in a hospital environment. As such, current trends in hospital information systems (HISs) must be monitored and if applicable, incorporated into the design of MIRG's clinical decision support systems.

Two key ideas that will be apparent throughout this thesis are *integration* and *standardization*. The ultimate goal is to design an architecture that will allow the integrated use of MIRG's ANN and CBR, and other potential clinical decision support systems, in a turn-key manner. By necessity this goal requires that a standards-compliant medical repository structure be created to support multiple uses of data. It is interesting to note that two core challenges currently facing the world of health information systems are also integration and standardization [Kuhn & Giuse 2000]. Key examples are the integration of hospital institutions into 'decentralized networks of health care delivery' and the standardization efforts required for the establishment of Electronic Patient Records (EPRs).

Kuhn and Guise also report that the Health care Information and Management Systems Society (HIMSS) 2000 survey of 858 health care providers showed the following application categories as being the most important to medical informatics professional and health care workers:

- Clinical Information Systems: Hospitals are moving from administrative-based data, to focus more on the clinical perspective.
- Web-based Applications: Web-based access to clinical repositories is employed in a growing number of systems. The web has the potential to support an advanced health care environment, offering secure health networks and new applications and services. Primary issues of concern are data security and confidentiality.
- Clinical Data Repository: As more clinical data becomes available, multiple use of this data is becoming increasingly widespread. Data is being reused and data mining techniques are increasing.

The framework's architectural goals and proposed system architecture, described in chapter 4 and 5 respectively, address all three of these current areas of interest to health care providers.

1.2 Thesis Objectives

The direction of the research undertaken in this thesis is predicated by a belief that if Clinical Decision Support Systems (CDSSs) are to have an appreciable positive impact on patient care, they must be standards compliant and promote inter-operability between diverse and distributed clinical applications and medical data. This implies that their support will be effective in multiple clinical settings rather than each institution developing CDSSs specific to their own isolated ‘silos of information’. Adhering to these principles would foster CDSSs that generate shareable and reproducible results in a wide range of hospital settings. The four main objectives of this thesis are listed below:

1. The main objective of this thesis is to design a medical information framework that will allow for seamless integration of various medical applications, including CDSSs such as the Case-Based Reasoner (CBR) and Artificial Neural Networks (ANNs), developed by the Medical IDEAS Research Group (MIRG). The framework will define concepts, ideas, organisational structures and architectural components that are considered most relative to achieving this vision. Jacobson concisely defines a framework as: *“a micro-architecture that provides an incomplete template for systems within a specified domain”* [Jacobson *et al.* 1999]. The framework’s architecture is designed to be flexible and scalable, in order to accommodate the initial medical domain *use cases* as well as future CDSS goals and long-term medical directions. A *use case* is a computer term used in the Unified Modeling Language (discussed in detail in future sections) for representing interactions between users and the system; the complete set of use cases describes the system functionality. For example, a future goal is to link MIRG’s software to the Children’s Hospital of Eastern Ontario’s Neonatal Intensive Care Unit real-time data repository. Ideally, the real-time data received from the data repository will be able

to be processed through ANN, CBR or other clinical decision support systems dependent upon the physicians' preferences. This greatly increases the need to extend the applicability of our medical data through strict conformance to accepted medical standards. The framework serves as a foundation to bridge this and other communication impediments caused by a lack of conformance with medical standards. To be used effectively the medical standards themselves must be positioned to fit within an architectural framework; suggesting and defining such an architecture is also targeted in this thesis.

2. An integral component of realising this work is to develop the above framework's detailed medical information goals. These requirements were not defined prior to conception of this thesis, but will be developed based on this framework's vision of an ideal CDSS and medical application environment built on a basis of shared and interoperable data. This vision will be founded on physician input and by expanding on current medical informatics and information technology architectures.

3. A third thesis objective is to create a model of medical informatics use cases to serve as a basis of comparison between this framework's requirements and others described in the literature. These generic use cases will hopefully be employed as a business-level model for describing MIRG's clinical information system initiatives with medical informatics colleagues in open-source forums.

4. The final thesis objective is the implementation of a small-scale prototype application based on the architectural design, the purpose of which is to validate the proposed architecture and choice of enabling technologies.

1.3 Thesis Contributions

The thesis contributions are categorized as contributions to knowledge and practical contributions. While the practical contributions are left for discussion in chapter 6, the contributions to knowledge are listed below, ordered from most to least important.

- 1) Designed a medical informatics framework for integrating diverse Clinical Decision Support Systems (CDSSs) and for interoperable and unified medical information exchange in a web-based environment, based on standardized open access to medical metadata.
- 2) Developed the framework's unifying vision, based on physician input and hospital information system trends, of an open source medical architecture that integrates generic medical applications. Additionally, the framework's architecture is designed to support future information interchange and remote processing of real-time patient data received from the Children's Hospital of Eastern Ontario's Neonatal Intensive Care Unit (CHEO's NICU) online data repository and from other generic Hospital Information Systems (HISs).
- 3) Created a medical business model identifying generic and reusable medical informatics requirements (represented by UML use cases) in the health care sector by combining medical architectural goals presented in the literature with those of this thesis' framework. The purpose of this model is to place MIRG's current work in a global context within the medical open source community and to clearly illustrate future research directions.
- 4) Suggested and analyzed choice of XML as an enabling technology to achieve CDSS semantic and logical application integration. Demonstrated that the means in which XML has been added to the framework's fundamental interchange components is

unique in the field of medical informatics, where XML is primarily used as a method for representing and exchanging the Electronic Patient Record.

- 5) Formally presented the architectural model for the above framework based on the Rational Unified Process and the Unified Modeling Language.
- 6) Identified components of the framework's long-term architecture to be implemented as future work within the Medical IDEAS Research Group. These 'work packets' will be assigned as future student projects, the completion of which will result in the realisation of MIRG's vision of interoperable and integrated hybrid clinical decision support systems.
- 7) Thesis work resulted in paper entitled 'Design of a Health Care Architecture for Medical Data Interoperability and Application Integration' by Catley and Frize. Accepted for publication in the EMBS-BMES 2002 (IEEE Engineering in Medicine and Biology Society) annual conference.
- 8) Thesis work resulted in paper entitled 'Overcoming Medical Standardization Barriers with XML' by Catley and Frize. Accepted for publication in the 12th International Conference for Women Engineers and Scientists 2002.

1.4 Thesis Outline

To meet the objectives outlined above, this thesis has been divided into six chapters. Chapter 2 covers medical and technological background information. Due to the multi-disciplinary nature of the work, this chapter covers a broad range of topics and is intended to familiarise the reader with concepts that are explored and developed in subsequent sections.

Chapter 3 involves a literature review of the state of the art in medical informatics. Because the final framework incorporates diverse ideas and designs from the

field of health care and information technology, this chapter is also quite broad and organised by idea. Firstly, other architectural designs for integrating distributed data and medical applications are described. This is followed by an overview of medical standards efforts and a discussion of the principles of open source medical software development.

A statement of the engineering problem to be solved is described in chapter 4; this includes the development of the framework concepts and ideas, such as goals and architectural principles. Chapter 4 also analyses the state of the art in medical information architectures covered in chapter 3, in an attempt to show that the architecture being designed solves a significant problem in an original manner. Comparisons are made at the functional and technological level.

The framework's architectural design and the implementation of a prototype application are described in chapter 5. The framework's object-oriented architectural design is modeled using the Unified Modeling Language. Chapter 5 concludes with a technology validation based on the framework's prototype implementation.

Chapter 6 summarises the architectural design and prototype results, with an emphasis on the contributions to knowledge and the practical contributions made in this thesis. Suggestions for future work and possible framework extensions are included.

Chapter 2 Overview of Relevant Medical and Technological Concepts

The integration of information technology with health care offers unique opportunities for solving medical problems. This chapter will explore relevant concepts from the field of medical informatics and information technology. MIRG's clinical decision support system software will be described, followed by an overview of the core information technologies proposed in this thesis for creating a framework supporting interoperability and integration of medical applications and medical data.

2.1 MIRG's Medical Software and Application Environment

One of MIRG's goals is to provide broad functionality for its medical decision aid tools. As expressed by Frize: *"Most neural network and expert system developments in medicine, as decision-aid tools, have narrow applications"* [Frize et al. 1996]. Currently MIRG is working on improving the predictive successes of its tools based on comparisons with clinical data and physicians' input with the view that: *"The application of case-based reasoning and neural network techniques to medical databases can be powerful techniques to aid physicians in making decisions about the management of their patients in various types of medical units"* [Frize et al. 1996]. It follows that MIRG's future goal is to provide flexible tools that can support the processing of external databases with minimal manipulations.

The diversification of these tools to support broader applicability is an extensive two-part endeavour. The first phase involves the creation and continuous improvement of the CDSS tools to provide optimum outcomes for different medical databases. This is an

ongoing and highly successful effort that has spanned the last ten years. MIRG is currently applying its ANN and CBR technology to diverse medical databases, such as adult and neonatal intensive care unit data, and coronary surgery patient data. Phase two, which forms the heart of this thesis, entails the creation of a framework to support data interoperability and application integration to sustain the broad applicability previously described by Frize.

The following sections provide a brief introduction to ANNs and CBRs. Additionally, the application environment in which MIRG's CDSS tools currently operate is described in order to better position the obstacles to be overcome before an integrated ANN and CBR system can be realized.

2.1.1 Artificial Neural Networks

In conventional von Neumann architecture, the Central Processing Unit processes logical instructions sequentially. In order to process data in such a manner researchers must first create a set of logical rules. However, this is not always possible when relations between the data are not known, as is often the case with medical databases. The complex relationships between medical data lend themselves well to processing by neural networks: *"Because of their non-linear modeling capabilities, ANNs have been extensively applied to non-linear statistical modeling problems and can be used to perform very complex recognition tasks. As a result they are natural choices for modeling complex medical problems when large databases of relevant medical information are available"* [Frize et al. 1997].

A set of artificial neurons, connected in parallel, forms an artificial neural network. Ennett describes the functionality of such a network: *"Nodes have inputs where the known information about the situation is introduced. The inputs are then each*

multiplied by a corresponding weight. The values are then summed in each processing unit, and fed through a transfer function to scale the output to a value within a limited range” [Ennett 1999]. Each node has an output that will serve either as the input to the next layer, or if the final layer has been reached, will result in an output.

MIRG’s ANNs are trained using a ‘training set’ comprising 2/3 of the original data (including the data outputs) and tested on the remaining 1/3 to determine the ANN’s best Correct Classification Rate (CCR) or Average Squared Error (ASE). Currently MIRG researchers spend a large proportion of their time preparing the databases for processing by the neural network. This is common amongst neural net developers; Yale estimates that ‘massaging’ the training data consumes 80% of the time involved in training a neural net [Yale 1997]. While time consuming, pre-processing is a crucial first step in order to develop a neural net that can be deployed in a real-world medical environment.

MIRG’s ANN was developed using Matlab’s Neural Net ToolkitTM, as a base on which several additional capabilities have been built. The ANNs operate as feed-forward neural networks that use the hyperbolic tangent function with back-propagation and require that all variables be categorized and/or normalized to range between -1 and 1. In MIRG’s training environment the normalized data must then be converted to Excel where it is saved as a comma-delimited file. The file is then opened in wordpad and a search is performed to find all commas and replace them with spaces. The text file is then ready for ANN processing. This is a time-consuming and error-prone process that would benefit greatly from automation. Figure 2-A demonstrates the data flow for MIRG’s DECH database:

Figure 2-A Data Flow in Ann Pre-processing

MIRG's accepted practice for avoiding experimental bias in variable elimination is to perform an initial pass through the neural net using all the input variables. The ANN is then run with a weight-elimination algorithm enabled, allowing the researcher to remove variables whose weights have very small values, with little or no impact on estimating the outcomes. At this point a new database has effectively been created, and the results obtained must be examined to verify that the CCR and/or ASE values are superior to those obtained in the previous pass. A researcher can create many different databases before developing one that produces optimum results and one that would be effective in a clinical setting.

ANNs are trained with large datasets involving thousands of patients and numerous input variables. It has been MIRG's experience that this processing is very CPU intensive and time-consuming, although it has benefited from recent innovations such as automating the adjustment of the 9 parameters that control ANN performance [Frize *et al.* 2000]. The important point to realise is that MIRG's trained ANNs have not yet been deployed in a clinical setting such as that described in this thesis (where ANNs are projected for use in the NICU). For this reason, specifics regarding the processing

speed of ‘trained’ ANNs (e.g. CDSS processing is only required for one patient rather than for thousands) are still unknown. However, work is currently underway to integrate the ANN into the NICU and deployment is expected within months, which will provide all the required information for future clinical implementation of this framework.

2.1.2 Case-Based Reasoning

Case-based reasoning (CBR) systems are used to provide physicians with diagnostic support. The underlying idea is that CBR technology provides analogy-based solutions and diagnoses to clinical problems by manipulating knowledge derived from similar, previously experienced situations, called cases [Manickam & Abidi 2001]. A new problem can then be solved based on previous cases and the physician’s domain knowledge. The assumption is that similar problems have similar solutions [Schmidt *et al.* 2001]. The most difficult part in deploying CBR systems is adapting the cases and their solutions to meet the user’s current needs [Taylor *et al.* 1993]. MIRG’s current CBR system does not modify the solutions of former cases to fit the current one. The rationale is that, while physicians are interested in obtaining knowledge about past cases for comparison purposes, they wish to reason about current cases themselves. Most CBR systems employed in the medical domain focus on the information retrieval aspect and allow the physicians to do the ‘actual thinking’.

By necessity, implementing a CBR system requires a large volume of current and correct clinical cases. MIRG’s CBR system is based on a modified Rete-matching algorithm to develop a match score based on a variable’s attribute values, whether it is string, word, character or number matching [Frize *et al.* 1997]. Additionally, MIRG’s CBR allows physicians to customise the match weights of different search variables for a

new patient case. MIRG's CBR tools are also projected for eventual deployment in the NICU.

2.1.3 CHEO's NICU Real-time Data Repository

As suggested by Ball: *"Clinical decision support systems are being used to enhance decision making and improve efficiency in diverse health care environments, from acute care to ambulatory practice. As clinical decision support systems become more commonplace, they will feed the data repositories that are key to evidence-based medicine"* [Ball *et al.* 2001]. An excellent example is the data repository being implemented at the Children's Hospital of Eastern Ontario (CHEO). Beginning in August 2002, CHEO will be deploying a real-time data repository in their NICU. This repository will function as an information management system, responsible for collecting patient device data, as well as data organisation and analysis. Real-time information from the NICU devices will feed the repository. It is probable that the repository will be remotely accessible to enable the sharing and transfer of data to secure external locales, such as other Hospital Information System (HIS) units, and to allow remote-patient monitoring by physicians. Similar initiatives have been advocated by the Canadian Institute for Health Information (CIHI); Marcoux (CIHI's applications development manager) claims that *"secure, real-time data transfer using open standards is key to the advancement of health information integration"* [Smith 2001].

Based on the successes of MIRG's CBR and ANN training, the decision was made to incorporate these tools into the NICU to interface with the real-time repository. Ideally, the ANN and CBR will provide an integrated solution that will be available to the physician in real-time. Research will also be carried out to develop means in which integrated CDSSs can work in a hybrid manner (where results from one CDSS feed the

input of another) to provide more accurate patient outcome predictions. This framework will extend the objective of local clinical decision support implementation to target wide scale distributed access to CDSSs. Future sections will discuss the benefits and challenges of creating this distributed solution.

Real-time NICU CDSS data processing has the potential to be very unique; currently most decision support used in hospital environments provide knowledge or handle administrative work; little work has been accomplished in terms of actual decision making. For instance, the majority of published ANN medical research is based on showing that the ANN can obtain a certain specificity and/or sensitivity in a research setting; however, actual deployment and usage in real-time clinical settings are limited.

2.1.4 Challenges Faced in the Current Environment

As described, extracting meaningful outputs from medical data processed via ANN technology is a dynamic process where the researcher is constantly testing new variable permutations and classification (categorisation) schemes in an attempt to obtain optimal results.

One concern is that the metadata that describes specific data-set variables, created during the time-intensive pre-processing process, is not being stored in an accessible repository for discovery purposes within MIRG. This could have significant negative impacts in terms of being able to effectively reproduce ANN results. An architecture is required whereby variable metadata information can be shared and updated in a central location, where it could be accessed by the entire research group. Enabling MIRG researchers to access this metadata could greatly reduce pre-processing time, in part by allowing reuse (inheritance) of classifications accomplished by other researchers.

Variable metadata could be safely published without compromising patient data confidentiality.

A further issue is that currently MIRG's acquired databases do not rigorously conform to medical terminology standards, making the metadata difficult to share in broader contexts, such as with other clinical decision support system applications. This lack of standardization is an impediment to preparing data for processing, as the researcher must first understand the meaning of all institution-specific variable encodings. As more medical institutions incorporate medical data standardization efforts, it is hoped that terminology servers will increasingly be used to describe medical data. As pointed out by Manickam, preparing the large number of clinical case data for CBR usage requires *"the terminological and conceptual standardization of heterogeneous clinical cases to a common information standard"* [Manickam & Abidi 2001].

Due to the fact that MIRG's ANNs and CBRs use different proprietary database products for managing data, it is not currently possible for data to be shared between these applications or with other researchers and applications over the Internet. What is required is a mechanism for sharing data between applications as well as between individuals, and forcing this data to conform to accepted standards, in order to allow meaningful sharing of variable metadata. Such a capability is offered through the use of the eXtended Markup Language.

2.2 eXtended Markup Language (XML)

From a health care perspective the eXtended Markup Language (XML) has an important role to play in promoting the acceptance of computer applications within the medical community. XML provides a valuable means for exchanging data in a flexible and efficient way; this capability will be exploited as the need for integration in health

care organisations of information and knowledge from distributed heterogeneous sources increases. In fact, XML has become an integral component in many clinical systems now under development [Okstein 1999], and while the number of applications currently being developed based on XML technology is already large, this number is expected to increase dramatically in the near future [Stalidis *et al.* 2001]. This thesis will explore the use of XML as a means of solving information sharing and interoperability challenges to form the basis for a future web-enabled system that supports CDSS application integration.

In order to understand XML's benefits it is important to have a realistic understanding of which system challenges this technology truly solves. This can be difficult because XML is a very flexible standard and it can be hard to grasp what it actually accomplishes [Lear 1999]. In simplest terms, XML enables data portability. As the use of Internet technology in health care becomes more prevalent [Anderson *et al.* 2000, Kahn & de la Cruz 1998], this is not a capability to be viewed lightly. Widom states that: "*The exciting promise of XML is that it turns the web into a global database*" [Widom 1999]. MIRG's goal is to capitalise on this promise by sharing medical data and enterprise applications, such as clinical decision support systems, with other medical research groups and health care facilities.

The subsequent sections will give an overview of XML, followed by a explanation of how this technology is applicable in the context of this thesis.

2.2.1 Overview of the eXtended Markup Language

The Hyper Text Markup Language (HTML) is currently the predominant format for information exchange over the Internet. However, as web applications become more sophisticated, some obvious faults are revealed, and HTML's simplicity becomes a limitation. HTML is poorly suited for data sharing between diverse enterprise

applications and databases, such as the clinical decision support system applications supported by MIRG. Essentially, because HTML is presentation rather than semantic content oriented, it is not suitable for situations where applications, not people, interpret and process the data. Additionally, HTML does not permit the creation of new data tags to facilitate information exchange between organizations that share common data types.

To solve these and other problems the World Wide Web Consortium (W3C) developed the eXtended Markup Language family of standards, foremost among which was XML. Introduced in 1996, XML is a relatively new standard, however it was developed with strong industry support and already exerts a strong influence on internets and intranets [Seligman & Rosenthal 2001].

Basically, XML makes data portable by separating the presentation of the data from the data content; in addition users are able to define original content by creating new XML schemas or Document Type Definitions (DTDs). Schemas, or DTDs, are sent with the data (or referenced within the XML document) to describe how the receiving application will interpret the markup tags, thereby creating self-describing data. The key idea is that the platform-neutral XML tags do not use a proprietary format, but rather an open and self-describing one, with the goal that: *“XML will be to data portability what Java is to application portability”* [Lear 1999].

The schemas are interpreted and potentially validated by XML parsers and subsequently directed to enterprise applications. In this way XML enables application-to-application communication since it is possible to create code that enables applications to both emit and receive self-describing XML data. Pokorny expresses this well: *“XML data is generated by applications and it can be consumed by applications”* [Pokorny 2000]. In

this way, XML facilitates the development of smart agents, as applications interpret and respond to information intelligently. This can improve the long-term reusability and interoperability of participating applications.

As a concrete example consider a scenario where physicians at CHEO could send XML-enabled patient data to MIRG's Artificial Neural Network application over internets and/or intranets, and conversely MIRG could send an XML-enabled result back to the physician in real-time. This example illustrates one of XML's main strengths: its ability to simplify data interchange, because organisations rarely standardize on a given set of tools and often must resort to complex Electronic Data Interchange (EDI) formats. However, while XML will allow simpler data exchange both within and outside organisations, it must be remembered that there is not always a one-to-one mapping of the concepts and data objects represented as tag sets in XML. The issue of incompatible data models must be considered; this will be addressed in detail in section 3.3 (Medical Standardization Efforts).

XML's roots lie in the Standard Generalized Markup Language (SGML). SGML was designed by ISO (the International Organisation for Standardization) as a mechanism for separating a document's structure and presentation using a standardized markup language. Each SGML document is accompanied by a Document Type Definition (DTD) that defines valid tags for the document being created. In addition, the ISO defined a Document Style Semantics and Specification Language (DSSSL) that defines how to display the SGML document. The major drawback of SGML is its complexity and breadth, the standard is over 500 pages long, making it very difficult to learn and implement.

Recognising these limitations, the ISO realised that a direct migration of SGML to address content and structure separation on the Internet would not be appropriate. However, it was apparent that a mechanism was required whereby applications as well as humans could interpret and utilise the profusion of information available on the Internet. To meet these demands, the W3C first proposed XML, as a simplified and extensible subset of SGML designed specifically for the Internet [W3C XML 2000]. XML 1.0 became an official specification in February 1998, and since that time an entire family of XML languages has been developed to handle emerging needs. To address the entire breadth of these technologies is beyond the scope of this thesis; however those that are relevant to this work are described in the following sections.

2.2.2 XML Schema Language

Due to XML's extensible nature new tags can be defined to describe previously undefined concepts. Industry consortia and other groups have created their own DTDs and schemas tailored to the data exchange needs of specific fields [Lear 1999]. Currently, significant efforts are underway to create standardized sets of tags for health care [Okstein 1999, Sokolowski 1999]. The idea is that domain-specific schemas can be published in public repositories allowing users to create and share schemas with business partners, facilitating client and server side validation of XML documents.

The XML Schema Language was published as a candidate recommendation of the W3C in October 2000, and will form the basis for XML metadata definitions used in this thesis. XML schema is similar to a data definition language for a relational database, as it specifies the valid structures, constraints and data types for XML documents [Roy & Ramanujun 2001]. XML schemas offer more powerful features than the traditional method for describing the structure and content of XML documents using DTDs, such as

providing the rich data typing associated with most programming languages, as well as the capability for defining new data types.

In addition, both schemas and DTDs can be used as a basis of comparison to determine if an XML document's structure is well formed (e.g. conforms to the rules of XML, also known as document validation), and to identify errors in the data itself. Error identification is known as 'data validation' or 'schema validation' and is obviously essential in medical environments where the integrity of the data is crucial. The XML Schema Language also supports inheritance; this means that new schemas can be derived from existing schemas and it is possible to override defined features with new ones as required. Based on its extended functionality, XML schema provides the power to create robust and sophisticated XML applications.

2.2.3 eXtended Stylesheet Language Transforms (XSLT)

XSLT is a current W3C recommendation and potential standard that transforms XML content from one form into another for the purpose of promoting interoperability between vocabularies. Using XSLTs, XML data can be transformed into another XML document, an HTML page, a C++ header/source file, a text file or any other format for which the developer creates a matching XSL transform. These transformations are processed by XML processors, the most common of which is the Xalan XML processor from the Apache open-source project. The ability to transform XML documents into diverse forms will be essential in the technology validation section, where XSLT will be used to attempt to transform XML documents representing patient data into a CDSS-processable format based on the specific type of clinical decision support required.

Closely linked to the XML Schema language and XSLTs is XML Namespace. A Namespace serves as a unique identifier for a particular purpose, such as a vocabulary;

this eliminates recognition errors amongst standards. Simply put, a namespace defines the area in which a particular name means something; two objects may have the same name as long as they are not in the same namespace.

2.2.4 XML Parsers

An XML parser interprets an XML document so that it can be used by applications. The most common parser is the Xerces4J parser created by the Apache open-source project; it forms the basis for both IBM's XML Parser for Java (XML4J) and Sun's Java API for XML Parsing (JAXP). XML parsers are normally categorised as being validating or non-validating and as supporting either the Document Object Model (DOM) or the Simple API for XML (SAX).

The DOM is a recommendation of the W3C; it provides a common interface for manipulating document structures with the aim that DOM-compliant code should be interoperable between different parsers. DOM creates hierarchical tree structures that contain all the elements (nodes) in the document, represented as objects that can be manipulated. SAX parsers are event-driven rather than structure oriented. The parser alerts the application that an event (such as the start or end of a document or element) has been generated within the document. It is then up to the application code to handle these events. If there is no application code for a specific event, it is discarded by the parser.

The choice of parser is based largely on what the application is meant to do with the data. Applications that work with large bodies of data where speed is a requirement would be best suited to SAX, since it allows only the necessary events to be handled. Applications that need to parse and store all the information for later use are best suited to the DOM.

2.2.5 XML and Databases

XML and database technologies complement rather than compete with each other, XML can be important in bridging the gap between documents and databases, both of which can be represented in XML. Pokorny states that with the *“Increasingly data-oriented role that XML is playing, it is possible to view XML as a language for data modeling; a well-formed XML document is a database and its associated DTD is its schema”* [Pokorny 2000].

XML must address similar problems to those handled in the database arena, such as: data retrieval, transformations and integration. More specifically developers must decide on currently unresolved issues, such as:

- How will data be extracted from large XML documents?
- How will XML data be exchanged (by shipping XML documents or by shipping queries)?
- How will XML data be exchanged between communities using different but related ontologies?
- How will XML data from multiple XML sources be integrated?
- Will XML data be stored in its native format or transferred to a relational or object model? In the later instance XML is viewed more as computer middleware. Okstein advocates an approach where XML is used for *“Extracting information from database tables and transforming the data into a format that can be used by clinical applications”* [Okstein 1999]. Bourret concurs with this approach, stating, *“A large amount of today’s data is stored in relational databases. Therefore an essential part of applications that use XML as a data exchange format is transferring data between relational databases and XML documents”* [Bourret et al. 1999]. Bertino and Catania dispute this viewpoint, their opinion is that: *“XML-documents are application-relevant objects to be stored and managed by a Database Management System (DBMS)”* [Bertino & Catania 2001], however they acknowledge that work remains in terms of developing appropriate indexing techniques for XML documents.

At the time of conception of this work, commercial systems that stored XML data in its native format were in their infancy: *“Since XML databases represent a new*

technology there has been up to this point no concerted effort to develop specifications specifically for the market” [XMLDB 2001]. This prompted the decision of transferring XML data into relational format for storage purposes. Examining how this architecture could benefit from storing XML data in native format is left for future work.

2.2.6 In which situations is XML an applicable technology?

The flexibility of XML and its associated schemas are particularly advantageous in the health care industry where clinical data are stored in many different formats, making it difficult to use these data in distributed organisations. Reconciling such data to standardized schemas is an important step in integration of enterprise application systems using the Internet.

In general, *“medical databases are characterized by a high degree of complexity in terms of quantity of items, number of parameter values and data types (free text, categorical, numerical and other)”* [Dugas et al. 2001]. For this reason, the context of the term ‘database’ varies greatly within the medical field, describing anything from a simple flat file consisting of several data fields to a massive cancer registry with hundreds of tables. MIRG’s databases often comprise both clinical (textual) and scientific (numerical and categorical) elements; XML is applicable as a technology that can represent both forms. In fact, Dugas claims: *“XML’s ability to separate data structures from specific programs is a major step in the field of biomedical informatics”* [Dugas et al. 2001].

While XML provides a powerful syntactic framework, the language is not unduly complex, boasts a quick learning curve and widespread acceptance; its flexibility is particularly suited to rapidly evolving environments [Geissbuhler et al. 2001], making it a good candidate for development within MIRG, where new medical ideas are constantly

being conceived and implemented. While XML has an important role to play in solving standardization problems, it is not capable of solving all data sharing challenges. Table 2-A, adapted from [Seligman & Rosenthal 2001], concisely summarises the communication levels in which XML plays a role:

Issue	Can XML Help?
1. Geographic distribution: Data can be widely distributed geographically. Handled by standards such as HTTP, SOAP, CORBA.	The Simple Object Access Protocol (SOAP) is the current W3C standard for transmitting XML data. SOAP acts as the envelope and the XML data is the payload. XML data can also be transmitted using standard Hypertext Transfer Protocol (HTTP).
2. Heterogeneous data structures and languages: Different data structuring primitives and data manipulation languages. Handled by standards such as SQL/Call Level Interface or other middleware products.	XML provides neutral self-describing syntax for heterogeneous data structures. XML can be used for representing both relational data and hierarchical distributions.
3. Heterogeneous attribute representations and semantics: different representations of the same concept.	Provides a mechanism for attaching reference metadata that describes data representations. XML does not address the data semantics - that is the accompanying text (e.g. as you would find in a repository).
4. Heterogeneous Schemas: Systems can assemble the same information elements into many different structures. Addressed by UML models, XML schemas or SQL Data Definition Language statements (schemas which describe the SQL tables).	It is likely that schema diversity will continue to be a problem even if XML schemas achieve wide usage. Mechanisms are required to describe and share schemas. It is also true that while organisations are very unlikely to share SQL schemas since they contain internal and proprietary data, they will share XML schemas that are separate from any secure or proprietary data.
5. Object Identification	XML provides improvements in describing attribute representation and semantics.

Table 2-A Communication Levels at which XML can Support Interoperability

Chapter 3 Literature Review

As discussed in the introductory material, the Internet is playing an increasingly significant role in medicine, underlying a growing number of systems for medical records, knowledge sources, decision support and education [Kahn & de La Cruz 1998]. Currently, most Internet applications in the health care industry are used predominantly for knowledge diffusion, however some researchers have started ‘building prototypes not only for information sharing, but also for disease diagnosis and patient care’ [Anderson *et al.* 1997].

The thesis introduction outlined a major problem facing today’s medical community: the integration of distributed and networked medical applications for interoperability purposes across the Internet and within intranets. Medical informaticians are recognising the “*need for better integration, clearer separation of knowledge from applications, as well as the necessity to integrate the diversity of roles in a health care network*” [Geissbuhler *et al.* 2001]. This poses challenges on many levels, such as integrating diverse platforms and programming languages, different communication protocols, and medical data representations based on conflicting data models. As stated by Xu: “*An integration architecture should support integration of components that either manage information and data or provide services. Integration is considered at the interface, communication and knowledge levels*” [Xu *et al.* 2001].

This chapter begins with a review of current integration architectures from the literature. Section 3.2 presents an overview of health care standards and section 3.3 outlines the progress of open source principles within the field of medical informatics.

3.1 Architectures Facilitating Interoperability of Distributed Medical Data and Medical Applications

This section will describe diverse architectural solutions published in the literature that provide:

- Interoperability of medical data
- Interoperability (or integration) of medical services
- Information dissemination of medical data

The architectures are categorised based on their main functional business goals and not based upon the technologies used to achieve these goals. For example, although XML was chosen as a core technology for sharing and exchanging clinical information within this thesis architecture, not all of the following architectures will employ XML to realise this objective. To highlight this categorisation scheme, each project's functional goals will be illustrated by high-level business use cases. The compilation of these developed use cases in conjunction with this framework's goals will form the 'overall medical enterprise model' presented in chapter 4 (section 4.1.3).

Examining all architectural choices will lay a foundation for the critical analysis presented in chapter 4. In section 4.2, architectures described in section 3.1 will be used as a basis of comparison for the framework developed in this thesis, in terms of contrasting business use cases, architectural design and implementation technologies. In order to facilitate chapter 4's critical analysis, each of the following architectures will be given a project name; this name is not necessarily employed by the authors and is employed in this thesis for reference purposes only (names employed by the original authors are denoted with *).

It is insightful to note that enabling medical data interoperability for diverse business purposes, particularly via XML, is currently a very explosive topic in the

literature. This is evidenced by the fact that all related reference material is very recent, with few articles dating earlier than 1998. Interest in medical interoperability has expanded significantly since the conception of this thesis and is gaining industry as well as academic support, with giants such as SUN Microsystems becoming key advocates in the field. Examining how employing this framework will uniquely position MIRC to exploit these emerging technologies will be addressed in detail in section 4.3. The aim of section 3.1 is to review the most applicable interoperability architectures presented in the literature in order to contrast these architectures with the proposed architecture and to position extensions identified within this thesis.

While the heading: “Architectures facilitating the interoperability of distributed medical data and applications’ is very general, many of the systems described below only implement a subset of this broader objective. Each system’s main purpose is classified under a general *functional goal* and a specific business use case model is developed to illustrate the business specific behaviour. Use cases represent the functionality provided by a system and are modeled using the Unified Modeling Language (UML), a “*language for visualising, specifying, constructing, and documenting the artefacts of a software intensive system*” [Booch *et al.* 1998]. For readers unfamiliar with UML, appendix A provides an overview of relevant UML terms and diagrams used throughout this thesis.

3.1.1 Functional Goal #1: Integrating Distributed Information Sources

The three projects characterised under this goal are geared towards the sharing and dissemination of biomedical research and clinical data. While each project employs XML in its integration interface, there is little focus on standardized information exchange and reducing duplication of medical data. As with most projects described in section 3.1, the main source of information to be shared is the Electronic Patient Record.

3.1.1.1 The SynEx Project*

SynEx is a 2-year European Health Telematics project, which addresses the issues inherent in the provision and use of multimedia patient records across large enterprise-wide networks [CHIME 1999]. Xu presents an architectural solution to complete the SynEx integration platform [Xu *et al.* 2001], his main emphasis is on describing reusable and flexible components for creating integration platforms. In this instance, integration is defined as allowing clients (physicians) to query multiple information (Electronic Patient Record) sources. Two main integration interfaces are described:

- 1) The interface between the clients and the services provided by the integration platform, where each service is a request that interrogates different data sources. This is accomplished by the PILOT component, which uses a multi-agent approach. Agents are used to send sub-queries to the appropriate data sources and to retrieve responses. The PILOT generates queries and constructs responses based on the client's specific DTD. Because the PILOT is XML-enabled, it can be implemented on different integration platforms.
- 2) The interfacing between the agents and the data sources is accomplished by MEDIATOR components. In this architecture, mapping modules defined by domain experts provide semantic interoperability. Mediators are used to transform information between the source and target systems, and contain intermediate repositories (IR) for the storage of information during the transformation process.

Semantic interoperability is defined at the Intermediate Repository level by an attribute-value representation known as a mapping. Each mediator is associated with a unique mapping process for its individual database. The authors state that achieving semantic interoperability based on ontologies and the potential use of central XML metadata, rather than employing mapping modules, might improve the scalability of their solution to larger and more complex cases. Figure 3-A from [Xu *et al.* 2001] illustrates the multi-agent mediation architecture and figure 3-B presents a simple use case model for the system.

Figure 3-A SynEx Architecture [Xu *et al.* 2001]

Figure 3-B SynEx Use Case Diagram

3.1.1.2 The EDRN Project*

The Early Detection Research Network (EDRN), supported by the National Cancer Institute, consists of 30 different EDRN centres distributed across the United States, and addresses the issue of distributed information sources. Their goal is to develop an architecture for disseminating and sharing knowledge between these centres, the challenge being to "*establish an informatics enterprise supporting biomedical research in a multi-database, multi-disciplinary environment*" [Crichton *et al.* 2001]. In accomplishing this goal each centre will retain their geographic and heterogeneous

implementations and will not re-implement their underlying system data models. EDRN researchers preferred to work in a web-based environment and to preserve their legacy information systems.

Crichton's team identifies the generating and managing of metadata as key in terms of achieving interoperability between heterogeneous data systems. XML was chosen for its capabilities in defining and managing metadata. The primary architectural objectives are as follows:

- Individual data systems be encapsulated to hide uniqueness.
- Communications between distributed services use metadata for data interchange.
- Defining standardized data elements stored in a data dictionary.
- Scalability and extensibility.
- Providing a standard mechanism for exchanging data results across distributed services.
- Allowing systems using different data dictionaries to be integrated.

The system is designed as a distributed object architecture where each node exchanges data using XML over CORBA based on metadata definitions. The middleware component framework allowed the heterogeneous distributed data systems to be connected for interoperability without re-structuring their data models.

A profile server was implemented to manage metadata resource definitions (profiles) about the products contained in each node of the distributed information system. Each research centre's local database was accessed through the centre's product server that was responsible for translating the retrieved local data into the neutral format suitable for exchange between research centres. This common format, based on metadata describing common data elements in cancer research, was created as a joint effort with the National Cancer Institute. The product servers also translate XML queries into local queries for

their specific databases. Figure 3-C from [Crichton *et al.* 2001] illustrates the system architecture and figure 3-D presents the system's use case model.

Figure 3-C EDRN Architecture [Crichton *et al.* 2001]

Figure 3-D EDRN Use Case Diagram

3.1.1.3 Internet-Based 3D Radiotherapy Planning and Information System (IRIS)*

The project's aim is to establish an Internet-based medical information and tele-learning system to promote radiotherapy research [Luttgau & Bendl 2001]. IRIS operates as a client-server architecture that links together different radiotherapy support modules such as: dose calculation, knowledge-based systems for automatic planning in radiotherapy and a virtual radiotherapy simulator. The client side is implemented as Java applets running on an Internet browser and the server side runs as a distributed, heterogeneous network written in C++ and Java.

Common Object Request Broker Architecture (CORBA) interfaces were used to provide technical interoperability between the heterogeneous platforms and languages of these and possible external modules. Java-Java communication is implemented using Remote Method Invocation (RMI) and database connectivity is accomplished via JDBC. Because computing radiotherapy plans is time-consuming, parallel processing is used and implemented using Parallel Virtual Machine (PVM) software used in the construction of systems written in C, Java or Fortran.

It should be noted that IRIS can currently only read files generated by the Heidelberg VOXELPLAN system and by commercial versions of VIRTUOSO; there is no conformance to accepted standards for exchange of radiotherapy-related information between different information systems. Future plans include possible conformance with the Digital Imaging and COmmunication in Medicine (DICOM) standard. Figure 3-E from [Luttgau & Bendl 2001] illustrates the system architecture and figure 3-F presents the system's use case model.

Figure 3-E IRIS Architecture [Luttgau & Bendl 2001]

Figure 3-F IRIS Use Case Diagram

3.1.2 Functional Goal #2: Standardized Information Exchange within and between Hospital Information Systems

Traditionally, each of the many groups within a large hospital has created their own knowledge and terminologies, in effect isolating themselves by creating natural information sharing boundaries. As shared patient data is being increasingly proposed as a means to provide more effective patient care, HISs worldwide are developing architectures for shared information exchange within their own HIS and with other collaborating centres. This is one of the more common areas for XML usage and it would not be feasible to describe all the efforts currently underway. The following will describe some of the more advanced initiatives, with particular emphasis placed on what is meant by 'integration' in each context.

3.1.2.1 The Hospital Information System DbMap Project (DbMap)*

Karadimas's team proposes an architectural solution for integrating legacy medical databases and diverse medical software applications within a hospital information system, using Java Database Connectivity (JDBC) and a central metadata description file in XML [Karadimas *et al.* 2000]. The key architectural objectives addressed were:

- Sharing and reuse of clinical data stored in relational models.
- Easy maintenance of applications that depend on external schemas by creating central metadata that can be edited and shared across applications and platforms.
- Transparent database access, including remote access.
- Enabling run-time switching between different databases that share a conceptual model.

This model is concerned primarily with accessing (querying) and replicating medical records and core administrative data stored in legacy databases at various locations throughout the HIS. It can be assumed that the objective of creating replicated databases is to prevent interference with ongoing processing; however, the article focuses

on technological aspects and does not clearly define the scope or context in which this technology will be operating. XML is used for the mapping description file, thus making mappings sharable and reusable. Karadimas defines a *DbMap* layer between the databases and the client application that automatically generates standard SQL statements based on metadata information stored in XML format. Figure 3-G from [Karadimas *et al.* 2000] illustrates the system architecture and figure 3-H presents a simple use case model for the system.

Figure 3-G DBMAP Architecture [Karadimas *et al.* 2000]

Figure 3-H DbMAP Use Case Diagram

3.1.2.2 Universidad Politecnica's Medical Database Migration (UPMDM) Project

Pons' team addresses a similar goal; their focus is on creating an XML and web-based architecture to enable interoperability between the different data inputs in a hospital information system: the administrative records, electronic medical records and diagnostic information [Pons et al. 1999]. The architectural goals of the UPMDM project are described below:

- To allow local processing of the data (Note: This system does not process data, it displays data. In this context the term processing is used very loosely to describe taking manual action based on the displayed data).
- To provide bi-directional data flows between server and client.
- To enable real-time monitoring.
- To adapt data for compatibility with other systems.
- To allow for remote data access.

[Pons et al. 1999] states that the increased need for information sharing (which they define as the ability to query remote data) within and between medical facilities has

resulted in the proliferation of medical information exchange standards . These exchange standards normally comprise both a terminology standardization component and a proprietary data transfer protocol; CORBA is a common implementation tool. In this project, intra-hospital communication will be enabled using XML, allowing the transmission of highly structured data using the hypertext transfer protocol (HTTP) instead of through proprietary communication solutions. The authors claim the use of XML will allow users to exploit the large amount of work that has been devoted to the development of medical standards.

The project was based on the ‘migration’ of an electrocardiography database containing 600 patient records. A necessary first step involved the creation of an XML DTD, coined Biosignal Exchange Markup Language, and based on a simplified version of the CEN/TC251 European health care standard. This was followed by the creation of a web-based client-server architecture where query responses are encoded in XML and interpreted by the client using the XML Document Object Model (DOM). Java servlets and JDBC drivers form the basis of the server implementation. This system is based on storing XML files in the web server and managing these files using Java servlets. The architecture handles data exchange within a hospital Intranet and does not offer security for remote information access. Figure 3-I from [Pons *et al.* 1999] illustrates the XML-based web architecture and figure 3-J presents the system’s use case model.

Figure 3-I UPMDM Architecture [Pons *et al.* 1999]

Figure 3-J UPMDM Use Case Diagram

3.1.2.3 Integrated Medical Information Systems (IntMedInfS) Project

This effort focused on integrating different subsystems grouped under a generic container: the hospital information system (HIS). The point is made that current HISs are not integrated but comprise distinct subsystems, such as: the HIS for managing Electronic

Patient Records, prescription information, administrative data, etc.; Radiology Information Systems (RIS); and Picture Archiving and Communication Systems (PACS) for X-ray information [Ahn *et al.* 2001]. HIS and RIS are commonly based on the Health Level 7 standard (see section 3.3) and PACS systems normally conform to the Digital Imaging and Communication in Medicine (DICOM) standard (section 3.3). The main goal of this project is to support web-based data sharing, exchange and retrieval between these two standard formats by creating an integrated XML DTD for Hospital Information Systems, RIS and PACS.

XML documents for individual units are generated by manually extracting medical information from HL7 or DICOM messages and converting these to XML based on the integrated DTD. The integrated DTD is designed from a patient viewpoint by combining each patient's HIS, RIS and PACS information. This DTD becomes the XML database schema and is stored in an XMLStore database (a proprietary XML-based database management system for storing native XML documents). This is one of the few projects that employs a database that stores XML datatypes in their native format. A web interface has been created for the integrated information system allowing searches based on the patient's identification. The system's use case model is shown in figure 3-K.

Figure 3-K IntMedInfS Use Case Diagram

3.1.2.4 Geneva Hospital's HIS/Clinical Information Project (GenHos-CIS)

Switzerland's Geneva University Hospital's hospital-wide clinical information system is among the more broad integration efforts found in the literature. The goal is to provide an architectural and technical framework for the convergence and integration of the system components based on shared medical terminologies and a central repository for the controlled vocabulary [Geissbuhler *et al.* 2001]. The central repository is used as a means of separating knowledge from applications. This allows the development of shareable services using XML for the syntactic framework and HTTP for Web transfer. In this context 'shareable services' can be interpreted as allowing all the organizations within that particular hospital being able to interpret and display the same data. While it is clear that XML is being used as a data transportation mechanism, the interactions between and within the application and middleware layer are not described in detail.

The authors state that when standard clinical XML-based object models mature and get wider acceptance, these should become the target for the evolution path of their

HIS object model. Currently, the repository standards are decided within the group and are not shareable with other HISs. Figure 3-L from [Geissbuhler *et al.* 2001] illustrates the system architecture and figure 3-M presents a simple use case model for the system.

Figure 3-L GenHos-CIS Architecture [Geissbuhler *et al.* 2001]

Figure 3-M GenHos-CIS Use Case Diagram

3.1.3 Functional Goal #3: Interfacing between EPR(s) and a CDSS

Currently, the bulk of interoperability work being done in medical informatics involves XML-enabling the Electronic Patient Record (EPR). To this end, many different projects and vocabularies are being created and it would not be reasonable to describe them all. The literature review focuses on projects that have some form of ‘link’ between the EPR and a CDSS(s).

3.1.3.1 Medical College of Wisconsin’s Integration of Bayesian Networks and Structured Reporting (IntBNWStrucRep) Project

Kahn employed XML as a mechanism to mediate between components of the computer-based patient record in order to integrate web-based systems for structured data reporting with a probabilistic decision support system named BANTER (Bayesian Network Tutoring and Explanation) [Kahn & de la Cruz 1998], which uses Bayesian networks (BNs) as its knowledge model. BN graphs contain nodes, each of which represents a stochastic variable. The semantic type of each node is identified in a node definition file.

The team employed an existing BN for the diagnosis of acute abdominal pain that required a 12-element model for diagnosis and used XML to express the model’s structure. An XML document generated from SPIDER (Structured Platform-Independent Data Entry and Reporting) replaced the original BANTER node definition file. The XML file was then used to instantiate the BN’s nodes. To ensure that SPIDER and BANTER could communicate (i.e. standardization issue), all concepts in the XML file were mapped to terms in the Unified Medical Language System (UMLS) Knowledge Source Server (see section 3.4). Figure 3-N presents a simple use case model for the system.

Figure 3-N IntBNWStrucRep Use Case Diagram

3.1.3.2 University of Malaysia's Case-Based Reasoning Interaction with Electronic Patient Record Repositories (CBR-EPR) Project

Case-based reasoning (CBR) clinical diagnostic systems require a constant supply of current cases from clinical experts [Manickam & Abidi 2001]. In order to be broadly applicable it is preferable that these cases should originate from more than one medical site. This aim can be achieved by retrieving data from heterogeneous XML-based Electronic Patient Record (EPR) repositories accessible via the Internet. However, generic EPRs do not all adhere to common information standards. To handle this problem the authors present an architectural solution based on transforming EPRs to Operable CBR-oriented Cases (OCCs), a term coined by the authors, compliant with specific CBR-based medical systems.

It is assumed that the EPR descriptive framework is well defined, using Health Level 7 (see section 3.2.1) for the transactional logic context and employing XML for the document object. This means that the EPR must simply be parsed to extract the relevant patient data. Following extraction vocabulary standardization is accomplished using the

Unified Medical Language System (UMLS) MetaThesaurus (to eliminate the usage of synonymous terms to denote the same concept), and concept standardization is achieved via concept standardizers using medical concept-specific ontologies. The final result is an OCC that can be processed through the Case- Based Reasoning system to provide web-enabled diagnostic services.

Utilizing repository data available on the Internet to feed CBRs is an interesting idea; however, it is unrealistic at this stage to assume there are an adequate number of standards-compliant sites of high-integrity EPRs available for public access by researchers. These techniques would only be advanced if HISs were willing to make available such records as anonymous data. Figure 3-O from [Manickam & Abidi 2001] illustrates the system architecture and figure 3-P presents the system's use case model.

Figure 3-O CBR-EPR Architecture [Manickam & Abidi 2001]

Figure 3-P CBR-EPR Use Case Diagram

3.1.4 Functional Goal #4: Integration of Clinical Decision Support in a HIS

As can be seen from the previous projects, the focus in the literature is more on administrative sharing of knowledge than knowledge sharing for clinical and processing purposes. Muller expresses this thought: “*commercially available HIS often focus on administrative tasks and do not provide additional knowledge-based functionality. Their traditionally monolithic and closed software architecture encumbers integration of and interaction with external software modules*” [Muller *et al.* 2001]. The following project is unique in the literature in that it covers integrating CDSSs into HISs.

3.1.4.1 Integration of CDSS into two Hospital Information Systems (CDS-HIS)

The aim of this project is to develop methods and interfaces to integrate knowledge sources from two separate HISs implemented at three university hospitals: Münster University, Düsseldorf University and Marburg University. The authors state that integrating decision support directly into the HIS has been proven more effective than isolated CDSSs, which often are used infrequently by physicians [Muller *et al.*

2001]. Furthermore, it is beneficial if the decision aids are integrated directly into the EPR (e.g. the results from decision support are automatically added to the patient's computerised record). This architecture integrates decision support into the HIS, connecting the EPR to valid databases of medical knowledge.

Only one CDSS was integrated into the HIS: a proven stand-alone scoring system for acute abdominal pain was supplemented with a communication interface. A future goal is to integrate more complex forms of decision support for medical data processing, such as guideline engines or clinical algorithms, into the HIS. Although a clinical decision support systems is being integrated with the HIS, it is not being deployed in an architecture that supports integrated CDSS processing, such that CDSSs can communicate with each other and provide a consolidated (combined) output.

The architecture was based on using distributed, flexible, component-based knowledge sources. CORBA was chosen as the middleware to establish the client-server relationship between the server objects and to provide interoperability between applications on different machines in heterogeneous distributed environments. XML strings are used for structured data exchange between the client-server components (their rationale for this choice is not explained, and technical issues are not addressed in detail).

A Knowledge Based System's (KBS) Manager is added to retrieve information from the HIS database (containing EPR data) and to interface this information with the knowledge sources (their version of decision support). The KBS Manager can be invoked in one of two ways: manually or based on a specific trigger for medical alerts. The data access component then retrieves all relevant patient data from the HIS database and transfers this data to a structured XML string. DTDs, medical schemas or standardized

Figure 3-R CDS-HIS Use Case Diagram

3.2 Current Medical Standardization Efforts

This section will present an overview of health care-related standards bodies and initiatives. These initiatives are becoming significant as the process of health care becomes an increasingly distributed process, necessitating co-operation among diverse medical organisations in shared patient-care scenarios.

An important element in creating relevant clinical information involves the use of standards. The driving force behind standards efforts in health informatics has been the development of the Electronic Patient Record (EPR) - a life-long, computer-based health record for each health care user. Acceptance of CBR systems has been hampered by the lack of a common vocabulary and language (medical informatics standards). Brandt states that until health care practitioners collect and maintain data in a standard format, it is nearly impossible to link data from one site to another [Brandt 2000]. To this end, Health Level 7 (HL7) has set up a working group to define a set of standards applicable to EPRs; however there is much work remaining.

To date, the EPR has been focused almost entirely on being in a form that is human readable, and has ignored requirements for information processing [Sokolowski

1999]. Conversely, Sokolowski points out that health care information transaction standards, citing Health Level 7 and Electronic Data Interchange (EDI) Accredited Standards Committee (ASC) X12, focus entirely on machine interaction and ignore the structure and content of the information viewed from a human perspective. XML is presented as a mechanism for achieving a standard health care object model that is capable of being processed and understood by both applications and humans.

The health care environment has complex information processing needs that are met by diverse information systems. In lieu of widespread health care standards, data transfer between information systems requires interfaces, which can be time-consuming to create since systems use different platforms, operating systems, programming languages and data formats [Brandt 2000]. There are four main categories of health informatics standards for data interchange.

- 1) Vocabulary Standards: To establish common definitions for medical terms.
- 2) Structure and Content Standards: To describe the data elements to be included in EPRs as well as standardizing the attributes of each data field.
- 3) Messaging Standards: To facilitate the distribution of health care data between diverse/disparate computer systems.
- 4) Security Standards: To ensure that health information remains confidential and protected.

The following sections will explore developments in health care standardization, specifically: Health Level 7 (addresses structure & content as well as messaging standards), the Unified Medical Language System (covers vocabulary standards) and the proliferation of XML-based medical markup languages (addresses vocabulary and structure & content standards).

3.2.1 Health Level 7 (HL7)

Many of the accepted standards in health care are focused on the electronic exchange of administrative and financial data within the medical environment. For instance, Health Level 7 provides standards for electronic interchange of financial, administrative and clinical information among distributed information systems within the hospital environment. HL7 is of interest not because it will be directly employed in this thesis, but because it is widely deployed in North American HISs and future integration of this framework with HISs is a possibility.

HL7 is an ANSI-accredited Standards Development Organization operating within the health care arena. The term 'Level 7' refers to the highest level of the International Organisation for Standardisation (ISO) Reference Model for Open Systems Interconnection, the application level. HL7 is actively involved with XML; Version 3 will only use XML encoding. To date, HL7's main focus has been the creation of a Clinical Document Architecture (CDA), which was approved as an ANSI standard in November 2000. As described on the official HL7 site: "*The CDA provides an exchange model for clinical documents (such as discharge summaries and progress notes) - and brings the health care industry closer to the realization of an electronic medical record*" [HL7 2002].

While HL7 is becoming increasingly popular, particularly for communicating data between heterogeneous North American HISs, it is based on establishing secure transmission of administrative rather than clinical data. Additionally, it does not achieve true semantic medical interoperability because the messages have not been standardized [Huff *et al.* 1998]. To improve this lack of semantic standardization, HL7 is currently collaborating with the Unified Medical Language System, described in the next section.

Most recently, as of January 2002, HL7, in conjunction with SUN Microsystems, has adopted the open-standard Electronic Business eXtensible Markup Language (ebXML) Messaging Service Specifications (e.g. Simple Object Access Protocol).

It is worthwhile to note that HL7 is employing XML and the UMLS as enabling technologies. While MIRG is not currently employing HL7's message exchange standards, this is not an unreasonable assumption if future CDSS deployment involved integration with Hospital Information Systems. Adhering to the same open standards technologies will make this migration significantly easier.

3.2.2 Unified Medical Language System (UMLS)

Within the medical community there is as yet no single accepted vocabulary standard or coding system. Different vocabularies have been created to meet the specific needs of diverse fields. This plethora of competing medical vocabularies representing the same information is "*a major barrier to the integration of autonomously developed applications*" [Ingenerf *et al.* 2001]. However, it may be unreasonable to hope that international organizations will agree on one standard for each medical domain. For this reason the UMLS was developed by the United States' National Library of Medicine (NLM). The following excerpt is from the 2002 version of the NLM's UMLS Knowledge Source Documentation:

"The UMLS project is a long-term NLM research and development effort designed to facilitate the retrieval and integration of information from multiple machine-readable biomedical information sources. The sources of interest include: descriptions of the biomedical literature, clinical records, factual databanks, and knowledge-based systems. Major barriers to effective retrieval and integration of information from these sources include (1) the variety of vocabularies and classifications used in different sources and by different users and (2) the sheer number and wide distribution of potentially relevant information sources..." [UMLS 2002].

The driving idea is that medical vocabularies provided simply as passive classes are no longer sufficient to meet the needs of sophisticated medical applications such as CDSSs. The UMLS provides “*machine-readable terminological sources that cover and integrate most existing medical vocabularies*” [Ingenerf *et al.* 2001]. Currently, more than 1000 individuals and institutions (of which MIRG is now included) have licensed and are employing the UMLS Knowledge Sources in their medical applications [UMLS 2002].

It is important to understand that the UMLS is not a proposed standard; rather it is a cross-referenced collection of standards and knowledge sources that addresses the problems of overlapping medical terminologies. In total, there are three UMLS Knowledge Sources, described below:

- 1) Meta-thesaurus: Contains semantic information about biomedical concepts, their various names and the relationships. Interconnects standard medical vocabularies and comprises nearly 80000 concepts from 60 different vocabularies [Schulz & Hahn 2001].
- 2) Semantic Network: A network of general categories of semantic types to which all concepts in the meta-thesaurus have been assigned.
- 3) SPECIALIST Lexicon: Contains syntactic information about biomedical terms- used for searching, indexing and lexical processing.

It is worthwhile to note for future work that the UMLS is currently integrating HL7 concepts and is working to XML-enable its knowledge sources, thereby greatly increasing its potential for automatic integration with medical applications.

3.2.3 Medical XML Vocabularies

Many domains are capitalising on XML’s ability to define application specific data; the number of special purpose XML languages and standards in all domains are expanding rapidly with “*several hundred already adopted since XML 1.0 was released in February 1998*” [XML.org2 2002]. The field of medicine is no exception, although

medical XML markup that is actually published in public repositories remains rare.

Without a commitment to sharing and standardizing this data, the result will be a proliferation of vocabularies duplicating the same information. Okstein corroborates this point: *“the acceptance and widespread use of XML as a method for defining clinical content depends on the establishment of standard vocabularies so that health care organizations can exchange electronic information with one another”* [Okstein 1999].

While XSLTs can transform XML documents from one XML vocabulary to another, based on data dictionaries, if the vocabulary is originally created in accordance with accepted medical standards, using mechanisms such as the UMLS’s metathesaurus to standardize diverse vocabularies, there is no need to employ such transforms. Several XML-based innovations in medical markup are listed below:

- **Medical Markup Language:** Developed in 1996 based on SGML, Japan’s Medical Markup Language employs XML as the meta-language for standardized medical interchange within Japan (with the goal of being used internationally). The aim is to represent *“comprehensive medical information”* [Araki *et al.* 2000]. The authors point out that as competing markups are developed the need arises to transform one markup to another for interoperability purposes - the extensible Stylesheet Language (XSL) is recommended for this markup transformation.
- **Bioinformatic Sequence Markup Language (BSML):** BSML is an open XML standard that describes bioinformatic data. Developed in 1997, this is one of the earlier initiatives and was sponsored by the National Human Genome Research Institute (NHGRI) as an evolving public domain standard for the bioinformatics community [Cover 2001].
- **Chemical Markup Language (CML):** Developed in 1998, CML has a broad scope: *“covering disciplines from macromolecular sequences to inorganic molecules and quantum chemistry”* [Cover 2002]. Both the schemas and the DTDs are publicly available on CML’s web site, however neither are currently registered in either of the two major XML portals (registries) described in section 3.3.2

3.3 Open Source Development of Medical Systems

In today's medical software development environment the terms *open-source* and *medical software* are often contradictory. Contrary to the accepted practice of peer-review and information dissemination within the medical research community, the field of medical informatics has been dominated by proprietary vendor solutions. Until recently, little emphasis was put on the sharing and interoperability of medical system architectures, software or data as a means of improving the quality of medical informatics research. This is largely due to it being in the best interests of proprietary software vendors to attempt to be non-standardized in order to prevent migration to other products. This practice has forced many medical researchers to create 'home-grown' solutions that often reinvent the wheel.

Arguably, this has adversely affected the ability of current medical software to meet patient, researcher and clinician's needs. Although the Institute of Medicine's *To Err is Human* [Kohn *et al.* 2000] report advocates the use of CDSSs to reduce medical errors, Yackel states that anecdotal reports shared at conference meetings confirm the dissatisfaction of most informaticians, clinicians and administrators with currently available medical software [Yackel 2000].

Adopting an open source model has the potential to improve the reliability, usability, interoperability, maintainability and scalability of medical systems. Open source models, by their very nature, encourage the use of open standards so that all developers have access to the same technologies - this can have a tremendous impact on the ability of programs to share data and interoperate. Open source holds potential to solve interoperability problems related to medical records by adherence to semantic standards in the medical domain. The code produced through the open source

development process is also likely to be of higher quality and better suited to handle sophisticated, complex problems, since the programs are tested in different problem domains other than those originally considered.

3.3.1 What constitutes an Open Source System?

The emphasis on open source development is naturally leading to adhering to open-source principles in the development of this thesis' framework. The goal is to build the framework on a foundation of open technologies and open medical standards.

The definition of what constitutes an open source system must be addressed. There are two main elements to consider. Firstly, does the license include access to the source code? And secondly, can the code be redistributed and modified under the terms of open source licensing, such as that provided by the GNU General Public License? XML by its very nature is an open source technology, as its key purpose is to share information via published schemas that can be improved by other domain experts. Many open-source XML tools and XML-based code are also freely distributed and accessible. While MIRG will be sharing the metadata that describe its medical databases for interoperability and application integration purposes, the current decision is that the source code describing MIRG's clinical decision support tools will not be shared.

3.3.2 OASIS and 'xml.org'

The Organization for the Advancement of Structured Information Standards (OASIS) is a key international open-source advocate dedicated to supporting product-independent data and content interchange. OASIS promotes structured information standards such as XML. This is achieved through the development of conformance tests and interoperability specifications that make standards usable. In this way "*OASIS does*

not compete with, but rather builds upon and supplements the work done by standards bodies such as W3C (for XML)” [OASIS 2002].

In June 1999 OASIS created XML.org, a publicly accessible service for the purpose of *“minimising overlap and duplication in XML languages and XML standards initiatives by providing public access to XML information and XML Schemas”* [XML.org1 2002]. XML.org is known as an ‘XML portal’, where contributors from business domains can publish their XML schemas. As of May 2002, XML.org is slowly gaining support from the health care domain with six domains having made contributions, an example being the Massachusetts Institute of Technology’s Gastro Intestinal Markup Language. In terms of competition, Microsoft has created a comparable portal (www.biztalk.org), which is eliciting a similar user response. However, the concept of sharing schemas is still in its infancy, and judging from the proliferation of markup languages being published in the literature, the potential for tremendous portal growth exists.

Gregory expands on these possibilities noting that the ability of repositories to supply schemas to run-time processes may become a critical aspect of portal-based systems: *“Rather than being mere distribution points, repositories will need to be well-designed to meet the needs of run-time systems. We may find that the standard registry/repository interfaces currently being proposed by XML.org and ebXML will be critical in enabling a global network of trading communities”* [Gregory 2001].

3.3.3 The Open Source Software Health Care Alliance (OSHCA)

As can be seen from the specific initiatives in section 3.1 and the standardization initiatives in section 3.2, medical system developers are realizing the futility of working in isolation and focusing on ‘isolated silos of information’. This realization is resulting in

a growing open source community in health care. Most notably, the Open Source Software Health Care Alliance (OSHCA) is a community of health care and informatics industries that promotes and facilitates the open source software concept in health care.

The OSHCA defines the following elements of the open source approach as being motivational to promoting the development of open source software and systems in the health care sector [OSHCA 2001]:

- The development process is driven by user needs.
- The development focuses on producing usable results.
- The development process has an unprecedented flexibility, ability to react to changing needs, and adaptability to various environments.
- Open source promotes the rapid spread of innovation.
- Open source promotes the reuse of components, collaborative development, and resource sharing.
- Open source health products are accessible to developing countries.
- Open source principles enable business models that are innovative, stable, and sustainable.
- Open source supports innovation in policy making.
- The transparency of open source software facilitates peer review and better quality assurance.

Of particular interest to MIRG is the OSHCA's annual forum for presenting projects that are driven by user needs and social concerns. The forum presents an opportunity for these projects to be subjected to early peer review.

Obviously, there are challenges to implementing open source systems. Currently, there are few medical open-source projects that go beyond the prototype stage and these are often hindered by the proprietary mindset of those involved in medical software development. Another major hurdle is the lack of interdisciplinary researchers with knowledge in the health care field that are interested in applying their programming talent to the medical domain.

However, it is evident that adhering to an open source model and incorporating the principles outlined above will provide long-term prospects for greater success of this thesis' targeted objectives. Embracing an open source model in the medical domain, as in other areas, is likely to be supported to a greater degree in the future as medical practitioners come to understand the flexibility and scalability benefits associated with reduction of proprietary control by vendors, and the Total Cost of Ownership advantages of open source software solutions versus purchased software.

3.3.4 Interoperability of Open Source Systems

This discussion of open source system development would not be complete without a discussion of interprocess communication aspects in open source systems. Models have been developed to address this issue, such as the Open Systems Interconnection (OSI) Reference Model; the basis of this model is to provide standard conventions for representing and communicating information. One of the aims of this thesis is to use XML to represent standardized information, thus minimising the necessity of gateway interfaces required for communicating between different network architectures. The problem with the traditional gateway approach is that in order to interface between 'n' representation formats, $n(n-1)/2$ interfaces are required. Therefore while achieving interoperability in a small-scale environment of five distributed systems would only require 10 interfaces, a larger integration, as might occur within a HIS comprising 10 distributed systems, would require 45 interfaces. In a distributed web-based environment with hundreds of users, the gateway approach would quickly result in an unacceptable number of interfaces or gateways.

Conversely, in the OSI reference model the idea is that each distributed architecture is responsible for transforming data into the standardized format required for

information interchange, meaning that between 'n' distributed architectures only n interfaces are required. Applying these ideas will be significant later in this framework when connectivity issues and minimising expenses in distributed web-based systems are considered.

Chapter 4 Designing a Framework for Integrated Clinical Decision Support and Medical Data Interoperability

MIRG's vision of an ideal hospital environment is one where physicians would be able to make real-time queries to diverse and integrated clinical decision support systems, such as case-based reasoners (CBRs) and artificial neural networks (ANNs), to confirm diagnoses or to suggest possible diagnoses. An excellent example of an environment that would benefit from such functionality is the neonatal intensive care unit (NICU) described in section 2.1.3, where physicians are often faced with difficult decisions regarding the potential survival and outcomes of very premature infants. To incorporate medical decision support systems into their decision-making process, the Children's Hospital of Eastern Ontario will be implementing an online data repository in their NICU for receiving real-time patient data via sensory input from medical devices.

Currently, MIRG does not have a system to access, share and extract useful information from large amounts of medical data such as would be received from the above-described repository. Without such a framework, the ultimate goal of receiving real-time infant data, processing this data through MIRG's CDSSs and returning information to the physicians in real-time would be very difficult. It follows then that the focus of this thesis is the design of an *open information technology framework* to achieve this, and other, medical system objectives.

Chapter 4 lays the foundation for this framework. Section 4.1 begins with the statement of the problem to be solved; this is an elaboration of the thesis objectives

outlined in section 1.3. The problem statement is followed by the contextual modeling of the environment in which the framework will operate (4.1.1), which leads to the formation of general architectural goals and principles that the framework must address (4.1.2). Based on these identified functional goals, in addition to those gleaned from section 3.1's literature review, a generic business model of use cases that support CDSS integration and medical data interoperability is developed (4.1.3).

Section 4.2 presents an analysis of the literature discussed in section 3.1 emphasizing differences between the approaches taken in those initiatives and the framework developed in this thesis, in terms of both functional and technological objectives.

Chapter 4 concludes with a discussion of why it is worthwhile to solve this problem (4.3). The main question that must be answered is what added value this framework brings to medical informatics and health care in general, and to the MIRG environment in particular.

4.1 Problem Statement

This work is divided into two distinct but interrelated problems, which can be broadly categorized as the contributions to knowledge (the knowledge dimension) and the practical contributions (the practical dimension) offered by this thesis, described respectively below.

Knowledge Dimension

The knowledge dimension centers on developing a complete and generic framework to support data interoperability and integrated clinical decision support in a distributed environment using Internet and/or intranet-based connectivity and open source XML technology. It follows that the specific problem being addressed is that to

the author's knowledge (as was demonstrated in the literature review) no such framework currently exists.

Firstly, founded on MIRG's vision of a complete medical information framework to support CDSS integration, specific architectural requirements will be developed incorporating input from physicians and other MIRG researchers. Using these goals as a basis, a generic and viable framework for standardized medical interchanges and data interoperability within the medical community will be designed and modeled.

The medical information infrastructure must be built using standards-based, reusable and replaceable components to promote inter-operability between diverse and distributed clinical applications and medical data. For example, the Unified Medical Language Knowledge Source Server will be used to standardize medical terminology. The architectural design must not be based solely on functional and technological factors but must integrate architectural principles and strategies of open-source medical development.

Practical Dimension

To reiterate, based on the potential of the World Wide Web Consortium's (W3C) standard XML to achieve data interoperability and reduce data duplication, this technology is proposed as a basis for realising data interoperability and CDSS integration within this framework. The practical dimension will address the problem of whether XML provides a suitable foundation for this framework. The applicability of the technology to the framework's problem domain, as well as issues such as ease of development, will be evaluated through a small-scale technology validation.

4.1.1 Contextual Scope of the CDSS Framework

The framework's contextual scope ('the big picture') was defined based on Zachman's Framework for Information System Architectures (see Appendix B for an overview matrix of the Zachman framework). The point of employing the Zachman framework is to provide a common context for understanding the system being designed, thus facilitating communication among system developers. In the first row of the Zachman framework requirements are identified in as generic a way as possible before consideration of the narrower technical aspects. This is useful for creating a system that will accommodate more sophisticated applications at a later date.

Row 1 of the Zachman framework identifies the scope of the system in question, as well as showing the context in which users will interface with the system. This is characterized as the What? How? Where? Who? When? And Why? of the enterprise and is illustrated in table 4-A. Note that the scope is identified for the complete interoperable and integrated framework being developed in this thesis, comprising all current and anticipated future contexts. This is essential, because while many of the articles in the literature focus on describing technological advances in medical informatics, there is a tendency to ignore the contextual and conceptual environment in which these technologies are operating. The purpose of employing the Zachman framework is to achieve consensus at the higher levels before implementing the lower technology level.

Scope	Details
Data (What?)	<ul style="list-style-type: none"> • Children Injury databases • Coronary Surgery Patient databases • NICU (Neonatal Intensive Care Unit) • AICU (Adult Intensive Care Unit) • Perinatal (CHEO) • Obstetrics (OH-GC)
Function How? List of Processes the Business Performs	<ul style="list-style-type: none"> • Artificial Neural Net Processing • Case-Based Reasoning Processing • Replacing Missing Values (current PhD work within MIRG) • Thermography Analysis (based on digital images) • Statistical Analysis
Network Where? List of Locations in which the Business Operates	<ul style="list-style-type: none"> • Carleton University, Ottawa, ON, Canada • University of Ottawa, Ottawa, ON, Canada • Children's Hospital of Eastern Ontario, Ottawa, ON, Canada • Canadian Neonatal Network of 20 NICUs (future) • University of British Columbia (future) • Coventry, UK (future)
Time When? List of Events Significant to the Business	<ul style="list-style-type: none"> • Real-time update of online data from CHEO's NICU • Time constraints associated with data access and processing
Motivation Why? List of Business Goals and Strategies	<ul style="list-style-type: none"> • Reducing medical errors via point of care Clinical Decision Support (e.g. estimating medical outcomes) • Reducing duplication of medical data representing the same information • Defining minimal data sets • Avoiding working with 'isolated silos' of information by standardizing medical data • Web-enabling medical data for sharing purposes and to incorporate the web services paradigm in CDSSs • Creating a scalable framework to support future CDSS functionality

Table 4-A Zachman's Framework: Enterprise Scope

The above table clearly shows the plethora of medical databases, locations and processes that the framework hopes to address, making a scalable solution a necessity. As

an added complexity many processes are still in their infancy, for example utilizing automated thermographical support in CDSS. In particular the extensive list of network locations illustrate the global impact this system wishes to have and plays a key role in advocating a web-enabled solution. A pictorial representation of the network locations that MIRG's enterprise must address is shown in figure 4-A; currently there is no connectivity between these locales.

Figure 4-A MIRG Network Connectivity Goals

4.1.2 Architectural Goals and Principles

Based on Zachman's contextual framework, table 4-B's architectural goals were defined. Ideally, MIRG's CDSS framework will address all these goals.

Architectural Goals	
1.	<p>To promote application integration, particularly for clinical decision support systems, such that all integrated CDSSs can process the same data.</p> <ul style="list-style-type: none"> a) To facilitate automated pre-processing of all MIRG databases for the purpose of CDSS training. b) To create an 'integrated solution' containing output obtained from all CDSSs. c) To create an environment where 'seamless chaining' of CDSS applications is possible. Chaining applications requires a logical flow that would make this desirable; note that work on hybrid (multi-modal) CDSSs is in its infancy and is currently being investigated by MIRG.
2.	<p>To publish medical metadata in web-enabled registry/repository structures that describe MIRG databases for discovery purposes within the medical research community.</p> <ul style="list-style-type: none"> a) This functionality could be expanded to support agent repository queries.
3.	To adhere to an 'open' system solution.
4.	<p>To facilitate the sharing of medical data (via the Internet) by adhering to common and open medical information standards.</p> <ul style="list-style-type: none"> a) To provide remote, web-enabled access to allow geographically distributed MIRG researchers to access and query MIRG databases. b) To develop partnerships with other medical research groups to increase the sharing of medical data.
5.	<p>To allow remote web-based processing of interoperable medical data through MIRG's CDSSs.</p> <ul style="list-style-type: none"> a) Additionally, other research organizations should be able to benchmark MIRG's CDSSs by processing their standards-compliant medical data through MIRG applications. b) To allow the most current version of MIRG's CDSSs to be available to researchers.
6.	<p>To accommodate real-time data transfer and processing of data received from the NICU's real-time data repository (and from other HIS data repositories as opportunities are presented).</p> <ul style="list-style-type: none"> a) This includes functionality for generating physician-alerts based on CDSS output dependent on acceptable medical threshold values.
7.	<p>To provide a high level of security to ensure medical data confidentiality and integrity, particularly for the above-described NICU processing.</p> <p>Security implications relating to medical data are referenced in this thesis.</p>
8.	To deploy early prototypes in clinical settings in order to validate the approach and gain end-user support.

Table 4-B Architectural Goals for MIRG's Integrated CDSS Framework

Understandably, physical realisation of all framework objectives is not feasible in the time frame of this thesis. Therefore, it is crucial to ensure that components resulting from this work, including the architecture, are reusable and scaleable enough to be deployed and extended in future work. One of the means by which this will be achieved is to explicitly state what this framework's architectural principles are. This thesis defines the following principles (table 4-C) as being relevant in the broader context of medical information systems for data sharing and advocates that they form the basis of any medical 'integration' framework. These principles will be applied in chapter 5's architectural design.

Architectural Principles	
Principle	Rational
Reduce Integration Complexity in Medical Systems Promote System Flexibility and Scalability of Medical System Components	<ul style="list-style-type: none"> a) Decompose components along dimensions of concern for easy addition of features. In developing medical systems it is crucial to be able to replace components without replacing the whole system. b) To promote reuse, adhere to the principle of high cohesion within modules and low coupling between modules. c) Decouple the user interaction modules from the application logic. It is important to be able to continually improve the user interface to reflect physician preferences. d) Encapsulate domain specific hardware so the hardware can be changed without affecting the software. Essential since new medical devices can be constantly added to update the repository.
Provide Defined Authoritative Sources for Medical Data and metadata	<ul style="list-style-type: none"> a) Authorised data must be accessible and available for re-use by any entitled users. b) Information needs to be useful, usable and reliable. c) Supporting event-driven systems means that information needs to be validated at the source.
Adhering to Proven Standards and Technologies	<ul style="list-style-type: none"> a) Systems should give priority to technologies adhering to industry standards and open technologies- this allows flexibility and adaptability in product replacement. b) Adhere to standards in all relevant areas.

Providing Security, Confidentiality, Privacy and Protection of Medical Information	<ul style="list-style-type: none"> a) Safeguard client trust. b) Protect assets. c) Minimize improper use of data. d) Minimize security violations.
Achieving Robustness of Medical Information Systems	<ul style="list-style-type: none"> a) Architecture should be robust, responsive and reliable with appropriate redundancy to protect against system failure. b) Must support error detection through exception handling and develop a consistent strategy for dealing with error handling. c) For real-time CDSS processing must minimise downtime and failures for high availability. d) Provide means to analyse errors to make quantitative evaluation of system performance (audit log). e) Encapsulate diagnostic components.
Effective Management of Time Critical and Resource Intensive Medical Data	<ul style="list-style-type: none"> a) Separate processing for time-critical and non-time critical data. b) Create separate components to meet timing requirements. c) Minimize copying of data and data duplication. d) Advocate an event-driven rather than polled model of information delivery to reduce wasted time spent periodically querying devices Assigning real-time priorities to medical device interrupts will ensure most critical data is given highest priority. Deploy an event controller to manage real-time alerts.

Table 4-C Architectural Principles for Medical Information Systems

A fundamental trade-off in any design is that between high flexibility and low cost. In this architecture the decision was made to emphasise flexibility. This is a reasonable decision considering that this framework is only the first step towards achieving a complete integrated system, where yet unknown medical databases and applications will be incorporated in a turnkey fashion. Allowing the system to easily accommodate new technological advances and applications, a fundamental requirement in a fast-paced medical research environment, is crucial towards attaining these goals. Additionally, the framework architecture must be extensible in order to support distributed and integrated applications based on diverse platforms, programming languages and operating systems.

These applications are distributed services that will ultimately be offered to internal and external MIRG users via internets and/or intranets. The security requirements are similar in nature to those of the electronic commerce paradigm, where parties external to the organisation are given access to application services.

Although flexibility prevailed in the trade-off between high flexibility and reduced cost, realistically the architecture has to account for the fact that a driving goal of MIRG's health-care clients is to reduce cost. The design attempts to reduce system expenses at other levels by incorporating open source technologies.

4.1.3 Business Modeling of the Medical Domain

Having identified both MIRG's current and future medical goals (section 4.1.2) as well as generalizing those of the literature (section 3.1), it is now possible to postulate a full set of generic use cases for medical data interoperability and CDSS integration. This will provide a global context in which to share MIRG's goals in the research community by modeling where MIRG's objectives relate to those of other medical informatics researchers. The model will also provide a foundation for the critical analysis presented in the following section.

The aim of this model is to identify medical enterprise objectives that are broader in scope than those addressed in this thesis, but which possible future work could implement. Functional system objectives are identified by generic Unified Modeling Language (UML) use cases. The scope of the area being modeled includes all behaviours required to provide medical data interoperability and application integration. **The full set of use cases collectively account for all current and future medical behaviour that this framework is designed to support.** The use case titles were chosen to be as self-describing as possible; while use cases derived from chapter 3's literature review have

already been presented in section 3.1, those specific to this thesis' framework goals (e.g. use cases shown in orange) will be described in detail in section 5.1. As mentioned previously, Appendix A provides an overview of UML use case diagrams and other UML artefacts. In this medical model the use cases have been divided between three packages:

- Interoperability Support (figure 4-B)
- CDSS-Interactions (figure 4-C)
- Broad Web-based Functionality (figure 4-D)

These diagrams are shown in the following three figures respectively. The use cases shown in orange will be addressed by the framework architecture in chapter 5, use cases shown in blue are currently implemented by MIRG, and remaining use cases represent possible future work.

Figure 4-B Medical Business Domain: Interoperability Support Use Case Package

Figure 4-C Medical Business Domain: CDSS-Interactions

Figure 4-D Broad Web-based Functionality

Note: The use case labelled 'Web-based Mobile Agent Searching of Medical Repositories' is based on the functionality described in Xu *et al.*, 2001 and Manickam and Abidi *et al.*, 2001 where mobile search agents take advantage of standardized XML-based databases to facilitate retrieval of information from remote repositories based on employing metadata as the search keys.

4.2 Critical Analysis of Medical Integration Initiatives

The purpose of this section is to analyse the literature from chapter 3 to illustrate how the framework proposed in this thesis extends and improves upon existing medical

architectural solutions in terms of both functional and technological factors. Particular emphasis is placed on showing how the thesis' framework provides a generic and flexible architecture, supporting broad medical use cases. In terms of technological factors, this section describes instances in which the framework's architecture makes novel medical use of emerging web technologies. To the author's knowledge no comparable XML-based CDSS medical framework exists; existing applications are heavily oriented towards data queries and information exchange between distributed sources in administrative settings. There is no shared CDSS processing of standardized, XML-enabled data.

4.2.1 Comparison of Functional Goals

Two key aspects of this thesis, as evidenced by the title: "A Health care Framework for medical data **interoperability** and CDSS **integration**..." are integration and interoperability. Ingenorf claims: "*Interoperability between health care agents (including heterogeneous software applications) is a key feature in distributed shared care settings*" [Ingenerf *et al.* 2001]. However, it soon became clear from the literature that different groups have different interpretations of interoperability, ranging from semantic medical language interoperability to the technical integration of medical applications and services. This thesis uses the IEEE definitions of interoperability (see table 4-D), with the extension of the term **integration**, meaning that both semantic and technical operability are present as well as a logical integration flow between modules. Interoperability can then be thought of as having the technical and semantic ability to share data, whereas integration extends these capabilities to actually use (process) the shared data between applications. An integration example is given by Manickam's system, where Electronic Patient Records are integrated in case-based reasoning modules.

<i>Interoperability:</i>	The ability of two or more systems or components to exchange information and to use the information that has been exchanged.
<i>Technical Interoperability:</i>	The transmission of data and the integration of services
<i>Semantic Interoperability:</i>	The actual usage of information (taking into account the context of usage)

Table 4-D IEEE Interoperability Definitions

As a means of comparing the functionality of the framework presented in this thesis with those described in the literature (section 3.1), three sets of matrices are used, one for each of the three use case packages defined in the medical business model (section 4.1.3). The functional comparison matrices are shown on the following pages (see table 4-E, 4-F and 4-G). The first column in each matrix represents MIRG's framework and items are 'checked' if they are part of the long-term solution that this framework is advocating. When reviewing these matrices it becomes apparent that many medical architectures have not harnessed the full potential in terms of logical integration that can be achieved through semantic and technical interoperability. It follows that the MIRG framework is designed to exploit full integration.

While the EPR is making inroads in terms of providing the physician with standardized, current patient information, little is being done with this information in terms of direct input of this information to CDSS processing. In other words, clinical support at this time does not normally extend to integrated decision support but focuses more on attempting to prevent misdiagnosis by providing the physician with the entire patient history. This is illustrated by the lack of 'checks' in the matrix representing CDSS-interactions and the relatively large number in the matrix representing broad web-based functionality. It can be concluded that in these systems XML is seen as a tool for

displaying data exclusively; XML's potential to achieve application integration based on its capability to promote semantic interoperability is being ignored. Another item of interest is that no single system architecture offers broad functional objectives; each has a very narrow focus. This aim of the thesis' framework is to join many of the narrower objectives of other systems into one unified framework.

General Use Case Functionality (From medical business model)	Project Team (Refer to Chapter 3)										
	MIRG Frame- work	SynEx	EDRN	IRIS	DBM AP	UP Data MIG	Int- Med- InfS	Gen- Hos- CIS	Int- BNW- Struc- Rep	CBR- EPR	CDS- HIS
Package Interoperability Support	✓		✓	✓		✓	✓	✓	✓	✓	
Standardized Medical Data	✓			✓		✓	✓	✓			
<i>Manual</i>	✓			✓		✓	✓	✓			
<i>Automatic</i>										✓	
<i>Based on Knowledge Sources</i>	✓								✓	✓	
Legacy Medical Database Migration to XML	✓				✓	✓					
<i>Automatic Migration</i>	✓				✓						
<i>Manual Migration</i>						✓					
Generate XML-based Medical Metadata (XML Schemas)	✓	✓	✓		✓		✓	✓			
<i>Automatic Generation</i>											
<i>Manual Generation</i>	✓	✓	✓				✓				
Publish Schemas in Medical Repository	✓				✓			✓			
<i>Access Metadata via Repository</i>	✓							✓			
Store Metadata in Network Nodes of Distributed System			✓								
Ensure Secure Access	Future Goal		✓	✓	✓						

Table 4-E Functional Comparison based on Medical Business Model's Interoperability Support Package (section 4.1.3)

Project Team (Refer to Chapter 3)											
General Use Case Functionality (From medical business model)	MIRG Frame- work	SynEx	EDRN	IRIS	DBM AP	UP Data MIG	Int- Med- InfS	Gen- Hos- CIS	Int- BNW- Struc- Rep	CBR- EPR	CDS- HIS
Package CDSS-Interactions	✓										
Invoke CDSS Processing & View Outputs	✓									✓	
<i>Remote web-based Invocation</i>	✓										
<i>CDSS Integration (Semantic Interoperability)</i>	✓										
<i>Hybrid CDSS Processing (Full technical and logical Interoperability- e.g. integration)</i>	✓										
Generate Interoperable/ Standardized Results	✓										
<i>Generate CDSS-Driven Alerts</i>	✓										
Integrate EPR and CDSSs	Future Goal								✓	✓	✓
<i>Update EPR with CDSS Results</i>	Future Goal										✓
<i>Feed CDSS with EPR Repository Data</i>	Future Goal									✓	
CDSS Training	✓										
<i>Semantic Interoperable Pre- processing</i>	✓										
Real-time Monitoring of Interoperable Patient Data	✓					✓					✓

General Use Case Functionality (From medical business model)	Project Team (Refer to Chapter 3)										
	MIRG Frame- work	SynEx	EDRN	IRIS	DBM AP	UP Data MIG	Int- Med- InfS	Gen- Hos- CIS	Int- BNW- Struc- Rep	CBR- EPR	CDS- HIS
<i>Package CDSS-Interactions</i>											
<i>Generate Device-Driven Alerts</i>	✓										✓
<i>Wireless Monitoring</i>	Future Goal										
<i>Monitoring from Within HIS (Intranet)</i>	✓					✓					
<i>Web-based Monitoring</i>	✓										
<i>Share Interoperable Data between HIS Modules</i>	Future Goal			✓		✓	✓	✓			
<i>Share Standardized Data between Multiple CDSSs</i>	✓			✓							
<i>Integrate CDSS into HIS</i>	Future Goal							✓			

Table 4-F Functional Comparison Based on Medical Business Model's CDSS Interactions (section 4.1.3)

General Use Case Functionality (From medical business model)	Project Team (Refer to Chapter 3)										
	MIRG Frame- work	SynEx	EDRN	IRIS	DBM AP	UP DataMIG	Int- Med- InfS	Gen- Hos- CIS	Int- BNW- Struc- Rep	CBR- EPR	CDS- HIS
Package Web-based Functionality											
Access to Distributed EPRs	Future Goal	✓			✓	✓	✓	✓		✓	
<i>Within HIS via Secure Intranet</i>					✓	✓		✓			
<i>From Remote Locations via (Secure) Web Connection</i>		✓					✓				
<i>Web-based Mobile Agent Searching of Medical Repositories</i>		✓									
Access Telelearning Resources	✓			✓							
Web-based Sharing of Biomedical Knowledge	✓		✓	✓		✓					

Table 4-G Functional Comparison Based on Medical Business Model's Web-based Functionality (section 4.1.3)

4.2.2 Comparison of Technological Enablers

The matrix presented in this section (see table 4-H) summarises the technical choices made in each of the architectures presented in section 3.1. Particular emphasis is placed on the means by which interoperability is achieved. For example, as mentioned in the previous section, XML is becoming a common semantic interoperability enabler in medical settings. The matrix shows that while XML is most commonly employed in isolation, it can also function in conjunction with CORBA, to transmit XML messages between distributed object nodes.

The matrices clearly demonstrate the lack of adherence to accepted medical standards and the strong reliance on proprietary and internal vocabularies. This lack of interest in standards is also shown by the means in which the systems are modeled, using adhoc mechanisms rather than employing standards such as the Unified Modeling Language (UML). UML is a widely adopted, formal industry standard and will be used throughout this work to model the framework's software architecture.

Technical Factors	Project Team (Refer to Chapter 3)										
	MIRG Frame- work	SynEx	EDRN	IRIS	DBM AP	UP DataMIG	Int- Med- InfS	Gen- Hos- CIS	Int- BNW- Struc- Rep	CBR- EPR	CDS- HIS
Standardization Mechanism											
<i>Proprietary</i>		✓	✓	✓	✓			✓			
<i>UMLS</i>	✓								✓	✓	
<i>HL7</i>							✓			✓	
<i>DICOM</i>							✓				
<i>CEN/TC251 (European Medical Standardization Body)</i>						✓					
Interoperability Enablers											
<i>XML-based</i>	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
<i>CORBA</i>			✓	✓							
<i>RMI</i>				✓							
Modeling Tools											
<i>UML</i>	✓										

Table 4-H Technological Comparisons

4.3 Problem Significance

The aim of the framework proposed in this thesis is to provide value and address current problems at two levels: within the realm of medical informatics in general, and within MIRG's application environment in particular. This section begins by examining informatics problems, categorised under the broader health care context of clinical decision support and medical data interchange, which this architecture is designed to resolve. This is followed by an analysis of how solving these issues will benefit MIRG's current working environment, as well as providing a basis on which more sophisticated MIRG applications will be developed, such as our collaboration with CHEO's NICU to provide effective clinical support in distributed clinical settings.

4.3.1 Significance From a Health care Perspective

To fully realise MIRG's goal of providing improved patient care through judicious use of clinical decision support systems, this work identifies two major health care problems that must be addressed:

- The lack of data unification and standardization in the current health care information infrastructure. This is commonly referred to as the 'silos of information' paradigm.
- The inability of the majority of health care infrastructures to promote sharing and interoperability between CDSSs.

The following expands on the above points and describes the benefits that could be achieved by a framework that addresses these issues.

4.3.1.1 The 'Silos of Information' Paradigm

While this 'age of information' has resulted in a profusion of data in many business domains, standardized, interoperable and complete data available for medical research is still lacking. Generally, data is collected by individual institutions (e.g. hospitals, general practitioners, pharmacies and insurance companies) and based on

institution-specific encodings. This lack of medical standardization and interoperability results in isolated "silos" of information, significantly decreasing the data's usefulness outside of its originating institution. To date, a significant portion of the work to standardize health care resources has been directed towards unifying administration and commerce within the hospital information system. These efforts obviously lack the medical dimensions needed for research.

It is important to realise that data unification is not a new problem, nor is it unique to health care. Data interoperability is relevant to any business application that wishes to employ data in a distributed manner, such as electronic commerce. However, it is only recently that health care has begun to migrate from the 'silos of information' paradigm to a 'knowledge' paradigm where medical information sharing amongst health care practitioners is becoming increasingly encouraged.

MIRG could serve to benefit from this knowledge culture in numerous ways, such as having access to increased volumes of medical data available for ANN training and receiving a broader knowledge base for our CBR tools. In turn, MIRG can make a significant contribution to the knowledge paradigm by:

- Ensuring that the knowledge obtained following processing by their clinical decision support tools does not result in another isolated silo of information but is integrated into the patient's broader history and available in a standardized format for future processing and research. A possible future goal is to integrate CDSS outputs directly into the Electronic Patient Record.
- Providing access to MIRG's CDSSs for benchmarking purposes within the research community

Reducing Medical Errors

MIRG's goal of providing clinical decision support is validated in light of recent studies indicating a high prevalence of medical errors in hospitals world-wide and

advocating CDSS as a potential means to help reduce these errors.

A landmark empirical study justifying the use of medical decision support is offered by the *Quality in Australian Health Care Study* conducted in 1995 [Wilson *et al.* 1995]. The nation-wide study of Australian hospitals reports that the majority of adverse events occurring in hospitals are the result of incorrect medical decision-making and diagnosis. An adverse event is defined as treatment resulting in disability or death. In total the study found that medical errors resulted in 16% of adverse events occurring annually that had significant costs in terms of bed years, increased disability and death. In the United States, Anderson roughly estimates that between 44000 and 98000 Americans die yearly as a result of medical errors, incurring an annual health care cost of \$37.6 billion [Anderson *et al.* 2000].

As a specific example, the 1997 UK Confidential Enquiry into Stillbirths and Deaths in Infancy found that 75% of babies who die during labor do so as a result of human error [UK 1996]. It has also been suggested that decision support could be very effective in reducing interventions such as Caesarean sections in situations where they are not warranted [Keith & Greene 2001].

It can be deduced that there is an obvious demand for clinical information systems that can reduce medical errors and reduce the costs associated with unnecessary procedures, litigation and long-term disability. This idea is corroborated by the recent Institute of Medicine Report on medical errors, which suggests employing clinical support systems to reduce medical errors and thereby indirectly cut costs [Yackel 2001].

The problem is that due to the ‘silos of information’ culture previously described, environments that would foster the development of CDSSs that would generate stable,

and precise results with data obtained from distributed medical centers are not available. For the most part, CDSSs are developed for a specific institution and the results are not generalisable to broader settings for benchmarking and peer-review purposes.

Therefore it can be postulated that the benefits obtained from interoperable CDSSs applied in an open-source medical environment could be enormous. The open-source paradigm in this context entails creating a means whereby international physician feedback is generated, collected and subsequent software improvements are made on a continual basis.

In addition, this work proposes that a framework that would enable interoperable CDSS processing would be scaleable to accommodate integrated processing of data between diverse CDSSs. Such a framework would facilitate the implementation of novel ideas such as hybrid reasoning, where one form of knowledge-based decision support is used to support another. As an example, CBR technology is well suited for integration with other CDSSs such as Rule Based Reasoning or Model-based Systems. A full investigation of the possibilities of hybrid (multi-modal) reasoning is left for future work.

4.3.2 Significance from the Medical IDEAS Research Group's Perspective

As the use of the Internet becomes ubiquitous, there has been a surge of information technology solutions that address data interoperability and standardization; in general, health care has so far failed to maximize these solutions. The proposed framework will position MIRG to leverage maximum benefit from these emerging technologies, with the objective of solving current developmental problems (discussed briefly in 2.1.4) such as:

- The absence of a technological means of providing standardized information exchange between MIRG researchers and with other

researchers, institutions and applications. In this context, standards refer to global medical concepts that are recognized by authoritative sources.

- The significant amount of time spent pre-processing data for neural network training and the lack of structure for sharing these ‘trained’ datasets between researchers.
- The lack of a structured framework for integrating MIRG’s CDSSs with emerging applications as MIRG tools evolve from the prototype to clinical validation stage. This can be expanded to include the integration of MIRG tools with each other in a hybrid manner as described above.
- The current inability of MIRG’s tools to receive and be trained using real-time data, which is crucial as the ‘shelf life’ of knowledge decreases and networks become tools for the most current medical data.

The framework will provide a flexible architecture to promote standardized information sharing via the Internet (and intranets). XML-enabled technologies will support transforming shared data into local representations for each of the integrated CDSSs. In this way, CDSSs may be added in a ‘turn-key ’ fashion with no impact on the data formats that are required by the system’s end users.

As mentioned in section 4.3.1 a further goal is to reduce ANN pre-processing time based on re-use and inheritance of XML schemas that represent the individual datasets. These schemas will be published and shared in standardized medical repositories. While this will require an initial time commitment in terms of group training and in formulating consensus on such schemas, it is hoped this will be compensated for in terms of increased system flexibility.

In terms of technology, the means in which XML has been added to the framework’s fundamental interchange components is unique. As can be gleaned from the functional matrices of section 4.2, XML is currently used more in EPRs and administrative support rather than for integrating medical information processing

systems. Validating that XML is an adequate technological enabler to meet the framework goals defined in table 4-B is another element of interest in this thesis and will be explored in section 5.4.

Chapter 5 Framework Architecture and Technology Validation

The purpose of this chapter is twofold. Firstly and most significantly, to model an architecture to meet the framework goals and principles outlined in section 4.1. A system's architecture comprises the most significant static and dynamic aspects of the system with the aim of providing a view of the whole design. The framework's architecture will be modeled and defined based on the software development process outlined in the Rational Unified Process (RUP) - RUP is "*use-case driven, architecture-centric, and iterative and incremental*" [Jacobson *et al.* 1999]. In this process different Unified Modeling Language (UML) artefacts are assigned to four main architecture models: the requirements model, analysis model, design model, and implementation model; subsequent sections will address the first three models.

After completion of the architectural modeling, this chapter's second goal is to validate the choice of enabling technologies for realising this framework through a small-scale prototype implementation of core model functionality (section 5.4). This prototype supports the framework's long-term objectives, and if successful will serve as a basis for a large-scale clinical implementation of the framework proposed in this thesis, involving full integration with the NICU's real-time data repository.

5.1 Requirements Model

The requirements model provides a foundation for the system design by capturing the user's needs in architecturally relevant use cases; the architecture must allow room for the implementation of all current and future medical domain use cases. In the

requirements model use-case modeling defines functional requirements; supplementary specifications are also included that describe additional requirements such as usability and performance.

5.1.1 Use Case Model: Actor Identification

The people identified within the Zachman framework, described in section 4.1, are candidate actors within the system. An actor in UML is not part of the system – they represent anyone (or anything) that interacts with the system [Quatrani 1998]. Actors were discovered based on populating the Zachman framework and including external devices and other systems with which the system must interact. This process is illustrated in Table 5-A:

Table 5-A Preliminary Actor Identification

Who will supply, use, or remove information? MIRG researchers, MIRG administrator, NICU administrator, NICU physicians (representing generic HIS users), external web users (researchers), MIRG CDSSs
Who is interested in a certain requirement? NICU physicians, MIRG Researchers, external web users
Who will support and maintain the system? MIRG administrator, NICU administrator
What are the system's external resources? UMLS Knowledge Sources
Will the system interact with any external hardware (H/W) or software (S/W) systems? S/W: NICU's data repository (as well as other HIS data repositories and medical databases) S/W: Potential alert systems (at NICU client side) H/W: Wireless devices for alert transmission H/W: NICU medical devices (cardiac monitors, blood pressure monitors, apnea monitors, pulse oximeters, etc.)

Based on the preliminary actor discovery it was possible to separate the responsibilities of the actors from those of the system. For instance, while the NICU real-time data repository was originally identified as a potential external software system, re-evaluating its functionality shows that the framework will need to modify repository data formats (see section 5.4) and it is thus possible to consider the repository as being contained within the system's whole. The following actors' roles were identified as shown in table 5-B:

Table 5-B Actor Roles

Actor	Role
Internal MIRG User (MIRG Researcher)	High clearance user who can view all medical schemas, publish updated schemas via the MIRG repository, and access current MIRG medical data and tools.
MIRG Administrator	Highest clearance user responsible for updating medical data, MIRG schemas and transforms, and CDSS tools, handling errors and maintaining site.
NICU Administrator	Highest clearance NICU user responsible for scheduling NICU devices for update in the real-time data repository.
External Web Users (representing affiliated research organisations)	Low clearance user who can view schemas and benchmark CDSS applications.
NICU Devices	Provide real-time or period data. Data is stored in NICU repository.
Alert Devices	Software/Hardware that receive interrupt-driven alerts based on CDSS results.

5.1.2 Use Case Model: Use Case Identification

Jacobson describes use cases as *“the functions the system provides to add value to the users; they can be considered drivers for the architecture”* [Jacobson 1999]. In the RUP use case identification is based on the following questions:

- What are the primary tasks the actor wants the system to perform?

- Will the actor create, store, change, remove, or read data in the system?
- Will the actor need to inform the system about sudden, external changes?
- Does the actor need to be informed about occurrences in the system?
- Will the actor perform a system start-up or shutdown?

Based on the above questions and the Zachman analysis, the framework's use case diagrams were created; use case diagrams show all use cases, actors and their relationships. In this architecture, use case diagrams are allocated to three separate use case packages:

- **MIRG CDSS Processing** (see section 5.1.2.1, figure 5-A)
- **MIRG Interoperability Support** (see section 5.1.2.2, figure 5-B)
- **NICU CDSS Support** (see section 5.1.2.3, figure 5-C)

The following sub-sections describe each package's use cases; use case names are emphasized in bold and numbered for increased clarity. For readers interested in the details of the use case flows, full descriptions are included. Readers interested only in a functional overview should focus on the heading entitled 'Brief Description'.

These use case diagrams represent a subset of the medical domain use case model created in section 4.1.3, and consist of use cases relevant to MIRG's clinical decision support system functional requirements for interacting with the NICU and other external web users. Note that many of the generic use cases from chapter 4's medical domain model are not included, specifically those involved in Hospital Information System (HIS) interactions. An example is updating the HIS's Electronic Patient Record with results from CDSS processing; while this is a possibility for future work it is currently considered beyond the scope of this framework's architecture. However, the presented use case diagrams provide a greater level of detail with regard to achieving medical data

interoperability and CDSS integration. The framework focuses on providing a backbone on which future work, such as that related to system integration with HIS(s), could be built. The goal is that eventually the use cases presented in this framework will be expanded to cover the full realm of medical CDSS-interactions and broad web-based use cases described in section 4.1.3's medical domain model.

5.1.2.1 MIRG CDSS Processing Package: Use Case Diagram and Descriptions

Figure 5-A MIRG CDSS Processing Use Case Diagram

5.1.2.1.1 Invoke CDSS Processing

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 1

1. Brief Description

This use case allows users to invoke CDSS data processing (ANN, CBR, etc.) from diverse locations through a secure connection.

2. Flow of Events

2.1 Basic Flow

- User invokes CDSS through graphical user interface, passing an XML document containing patient data to be processed and user parameters.
- System parses document to extract and store user parameter values and to store patient data in a medical audit log for error, recovery and auditing purposes.
- System invokes **ValidateData** to determine if the data being received conforms to its corresponding schema.
- [Data conforms] Based on identified CDSS user parameter value the system determines the specific transform that is required.
- System converts data into appropriate clinical decision support format based on invoking the **Transform Medical Data** use case (and passing the transform identifier).
- Once data is successfully transformed, **Invoke CDSS Processing** invokes **Process Medical Data by CDSS**.
- Upon receiving response from CDSS, received CDSS result is stored in MIRG's medical audit log along with its input data, for error, recovery and testing purposes.
- Application logic invokes **Generate Interoperable and Standardized Results**.
- Results are generated and displayed to user (client) through a graphical user interface.

2.2 Alternative Flows

2.2.1 < Data does not conform to its schema >

Inform user by displaying error message, CDSS processing suspended.

2.2.2 < Multiple CDSSs requested>

The possibility exists that user has invoked more than one CDSS either via **CDSS Semantic Interoperability** (user requires processing by multiple specified CDSSs) or **Hybrid CDSS Processing** (user requires logically integrated processing where the results from one CDSS affect the processing of the next, and processing proceeds in a logically sequential manner). Both use cases contain the same functionality as **Invoke CDSS Processing** with the distinction of requiring additional processing stages

and invoking **Generate Integrated Response** (rather than **Generate Interoperable and Standardized Results**).

3. Preconditions

3.1 User is registered (e.g. **Authenticate User and Establish Access Parameters** is successful).

4. Postconditions

4.1 CDSS output is displayed to user (or error message is generated).

5. Extension Points

CDSS Semantic Interoperability, Hybrid CDSS Processing, Validate Data, Transform Medical Data, Generate Interoperable and Standardized Results, Generate CDSS-driven Alerts

The use case logic for **Invoke CDSS Processing** is shown in the following activity diagram (see figure 5-B); note the potential parallel execution paths if more than one CDSS is invoked through **CDSS Semantic Interoperability**. Activity diagrams and use case diagram relationships are explained in more detail in Appendix A.

Figure 5-B Activity Diagram for Invoke CDSS Processing

5.1.2.1.2 CDSS Semantic Interoperability

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 2

1. Brief Description

This use case allows user to invoke processing by multiple CDSSs; based on having each CDSS process the same data (semantic data interoperability). This does not imply any logical associations between CDSS units, therefore processing could occur

concurrently if adequate processing resources were available. **Invoke CDSS Processing** is the parent use case of **CDSS Semantic Interoperability** (the child use case), meaning that the child inherits all the functionality of the parent (including <<extend>> and <<include>> relationships).

2. Flow of Events

2.1 Basic Flow

- User invokes CDSS processing requesting multiple CDSSs.
- Subsequent flow for validation and transformation is as in the parent use case with the modification that schemas and transforms must be retrieved for all pending CDSS processing.
- **CDSS Semantic Interoperability** invokes CDSSs for processing. If sufficient system resources exist this processing can be executed in parallel.
- Upon receiving response from CDSSs, system invokes **Generate Integrated Response** to create an XML-based response combining output from all invoked CDSSs.
- Results are generated and displayed to user through a Graphical User Interface.

3. Preconditions

3.1 User is registered (e.g. **Authenticate User and Establish Access Parameters** is successful).

4. Postconditions

4.1 CDSS output is displayed to user (or error message is generated).

5. Extensions

Hybrid CDSS Processing, Validate Data, Transform Medical Data, Generate Integrated Response, Generate CDSS-Driven Alerts.

5.1.2.1.3 Hybrid CDSS Processing

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 3

1. Brief Description

To invoke processing by multiple CDSSs, based on having each CDSS process the data in a logically meaningful sequence (integrated CDSS processing). This implies that there exists pre-established logical relationship among CDSSs of which the system logic has been made aware. **CDSS Semantic Interoperability** and **Invoke CDSS Processing** are the parent use cases of **Hybrid CDSS Processing**.

2. Flow of Events

2.1 Basic Flow

- User invokes CDSS requesting integrated processing by multiple CDSSs.

- Subsequent flow for validation and transformation is as in **CDSS Semantic Interoperability**.
- **Hybrid CDSS Processing** invokes CDSSs for processing based on a pre-established order for that particular CDSS set.
- Upon receiving response from CDSSs, the application logic invokes **Generate Integrated Response** to create a unified response based on the integrated output from hybrid CDSS.
- Results are generated and displayed to user through a Graphical User Interface.

3. Preconditions

3.1 User is registered (e.g. **Authenticate User and Establish Access Parameters** is successful).

3.2 Pre-established logical order exists for that combination of CDSSs.

4. Postconditions

4.1 CDSS output is displayed to user or error message is generated.

5. Extensions

Validate Data, Transform Medical Data, Generate Integrated Response, Generate CDSS-Driven Alerts.

5.1.2.1.4 User Registration

Use case package: MIRC CDSS Processing (figure 5-A) Use case number: 4

1. Brief Description

Prior to invoking clinical decision support a user must register with MIRC to establish their security level and user privileges.

2. Flow of Events

2.1 Basic Flow

- User initiates CDSS session by entering pre-established user information through MIRC's graphical registration interface.
- User information is validated via **Authenticate User and Establish Access Parameters**.

2.2 Alternative Flows

2.2.1 <Authentication fails >

Secure session is not established.

2.2.2 <New user registers>

Based on system security requirements user may have the ability to register at that time through registration interface. If this is the case further user information will be requested and user will be authenticated.

3. Preconditions

3.1 Pre-set list of registered users has been established.

4. Postconditions

4.1 Secure session is established and user may invoke MIRG functionality as per his/her privilege level.

4.2 If new user has registered, user security and privilege levels are updated.

5. Extensions

Authenticate User and Establish Access Parameters

5.1.2.1.5 Authenticate User and Establish Access Parameters

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 5

1. Brief Description

Before performing clinical decision support or allowing user to access secure system resources the application logic must authenticate the user and establish the user's access parameters. This use case could be accomplished using passwords or PKI-based authentication and would form the basis of future security implementations.

2. Flow of Events

2.1 Basic Flow

- This use case is invoked by **User Registration**.
- Application logic verifies user's security and access parameters.
- [User approved] Invoking use case is informed and secure session is established.

2.2 Alternative Flows

2.2.1 < User rejected>

Error message generated.

3. Preconditions

3.1 User has registered via the **Register User** use case.

3.2 **Authenticate User and Establish Access Parameters** implies that a pre-set list of secure users and privilege levels has been established against which users would be authenticated.

4. Postconditions

4.1 User may request clinical decision support or access resources.

5.1.2.1.6 Validate Data

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 6

1. Brief Description

Patient data (transferred as an XML document) must conform to its standardized schema to ensure data integrity prior to CDSS processing. This is essential because CDSSs require that data be in a rigid format to generate valid results.

Note: It is probable that data received from HIS repositories will include missing data values; missing values negatively impact ANN performance [Ennett *et al.* 2001]. Currently, hybrid CDSS research is being conducted within MIRG to discover methods for replacing these missing values with meaningful data inputs. When this research is complete the **Validate Data** use case will be extended (if required in a particular CDSS transaction) to include the **Replace Missing Values** use case. The **Replace Missing Values** use case is currently the subject of PhD research within MIRG and will not be discussed further in this thesis.

2. Flow of Events

2.1 Basic Flow

- **Invoke CDSS Processing** invokes **Validate Data**, passing the XML document to be validated and the desired output (from which the application logic can determine the corresponding schema identifier).
- Based on the schema identifier the appropriate schema is retrieved from MIRG's schema repository.
- Validation is performed by a parser that compares schema rules with the data stored in the XML document.
- If required (e.g. the data includes missing values and the requested CDSS is negatively impacted by the presence of missing values) **Replace Missing Values** is invoked.
- Validated document is returned to invoking use case.

2.2 Alternative Flows

2.2.1 < Schema is not found in the repository>

Error message generated.

2.2.2 < Document is not valid>

Parser generates an error (exception thrown) - processing does not proceed.

3. Preconditions

3.1 Schema must be stored in MIRG's medical schema repository.

4. Postconditions

4.1 Data ready for transformation via **Transform Medical Data** use case.

5. Extensions

Replace Missing Values

5.1.2.1.7 Transform Medical Data

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 7

1. Brief Description

Transforms standardized XML-based medical data into appropriate format for processing by desired CDSS(s).

2. Flow of Events

2.1 Basic Flow

- **Transform Medical Data** receives XML document and CDSS identifier(s) (from which the application logic can determine the corresponding transform identifier).
- Medical data is transformed into the correct format for CDSS processing.
- Processed data is returned to invoking use case.

2.2 Alternative Flows

- 2.2.1 < Appropriate transform not available (due to unknown CDSS identifier)>
Error message generated.

3. Preconditions

- 3.1 Data conforms to its schema.
- 3.2 Relevant transform is available.

4. Postconditions

- 4.1 Data ready for processing by CDSS.

5.1.2.1.8 Process Medical Data by CDSS

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 8

1. Brief Description

Invoke CDSS Processing passes transformed data to MIRG's CDSSs for processing. **Process Medical Data by CDSS** is an abstract use case that represents generalized clinical decision support system processing. While the actor perceives that he/she is interacting with **Process Medical Data by CDSS**, in reality specific concrete use cases are being instantiated as required (e.g. **CBR Processing** and/or **ANN Processing**). The rationale behind this choice is that the design must be expandable to include future CDSSs.

2. Flow of Events

2.1 Basic Flow

- **Invoke CDSS Processing** invokes **Process Medical Data by CDSS** passing the data to be processed.
- Processing is performed based on invoking required concrete use cases (e.g. one or many CDSSs).
- CDSS output is returned to invoking use case.

3. Preconditions

- 3.1 Data has been transformed into appropriate format based on **Transform Medical Data** use case.
- 3.2 Interfaces to CDSS(s) have been pre-established.

4. Postconditions

4.1 Data is ready to be transformed into standardized and interoperable result.

5. Extensions

CBR Processing, ANN Processing

5.1.2.1.9 Generate Interoperable and Standardized Results

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: 9

1. Brief Description

The textual and graphical data (and potentially other multi-media formats) generated by CDSS processing must be transformed into standardized XML format for transmission to the client. Relevant alerts must also be displayed as XML messages.

2. Flow of Events

2.1 Basic Flow

- **Invoke CDSS Processing** invokes **Generate Interoperable and Standardized Results** passing CDSS result(s) and the identifier of the CDSS(s) that generated this result(s).
- The CDSS result is compared with the session's patient threshold values (determined by the physician and stored locally at the server for the duration of the session), which specify the CDSS result's acceptable range(s) (see alternative 2.2.1).
- Based on the CDSS identifier the appropriate output transform is retrieved.
- Based on the obtained transform, textual and graphical data are processed into a combined XML 'result' document.

2.2 Alternative Flows

2.2.1 < Acceptable patient range has been exceeded>

Generate CDSS-Driven Alerts is invoked. The alert is included in the final interoperable and standardized XML-based result, based on combining the CDSS and alert transforms.

2.2.2 < Multiple CDSS identifiers are present>

The system combines transforms for both results, invoking **Generate Integrated Response**, which inherits all of the above functionality and provides extended XML-based transform processing.

3. Preconditions

3.1 CDSS threshold list is available.

3.2 Appropriate output transforms are available.

4. Postconditions

4.1 Combined XML-based response is generated and ready to be viewed by client.

5. Extensions

Generate CDSS-Driven Alerts, Generate Integrated Response

5.1.2.1.10 Generate Integrated Response

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: **10**

1. Brief Description

Generate Integrated Response extends the functionality of its ‘parent’ use case (**Generate Interoperable and Standardized Results**) to handle the case where output from multiple CDSSs was generated; this is accomplished via extended transform processing that generates an integrated XML-based response.

2. Flow of Events

2.1 Basic Flow

- As in parent use case; however, a unified response combining output from multiple CDSSs is created based on importing nested stylesheets for either **CDSS Semantic Interoperability** or **Hybrid CDSS Processing**. This implies extended XML-based processing capabilities that will be discussed in section 5.4.

3. Preconditions

- 3.1 CDSS threshold list is available.
- 3.2 Appropriate output transforms are available.

4. Postconditions

- 4.1 Unified XML-enabled output is generated.

5.1.2.1.11 Generate CDSS-Driven Alerts

Use case package: MIRG CDSS Processing (figure 5-A) Use case number: **11**

1. Brief Description

This is a simple extension use case that extends **Generate Interoperable and Standardized Results**.

2. Flow of Events

2.1 Basic Flow

- **Generate Interoperable and Standardized Results** invokes **Generate CDSS-Driven Alerts** passing the alert data.
- Based on the stored alert transforms an XML-based alert document is generated and returned to the invoking use case to be included in the final response.

3. Preconditions

- 3.1 CDSS result exceeds acceptable patient range (based on physician threshold values).

4. Postconditions

4.1 XML-based alert document is ready to be included in final result document that is transmitted to the client.

5.1.2.2 MIRG Interoperability Support Package: Use Case Diagram and Descriptions

Figure 5-C MIRG Interoperability Support

5.1.2.2.1 Update MIRG Medical Schema Repository

Use case package: MIRG Interoperability Support (figure 5-C) Use case number: 1

1. Brief Description

In order to share standardized information, authorised MIRG users may include new schemas, such as those obtained following ANN variable weight elimination, in the MIRG medical schema repository. Many schemas may be added to the repository; the only requirements are that each schema is valid and that no two schemas share the same name (or version number).

2. Flow of Events

2.1 Basic Flow

- User enters new schema information or updates an existing schema (e.g. creates a new schema version).
- Most current schema version is added to the repository.
- Schemas may be viewed by registered users.

3. Preconditions

3.1 Use case may only be invoked by an authenticated user with the appropriate privileges (e.g. MIRG Administrator or Internal MIRG User).

4. Postconditions

4.1 New schema is added to repository and future user requests can reference this schema (or updated schema has a new version number).

5. Extensions

Update MIRG HIS Schemas

5.1.2.2.2 Update MIRG HIS Schemas

Use case package: MIRG Interoperability Support (figure 5-C) Use case number: 2

1. Brief Description

CDSS processing is based on stored medical schemas that correspond to the relational repository structure of the medical data stored in the NICU (HIS). To ensure valid CDSS results, the integrity of these schemas is crucial; therefore they can only be upgraded during scheduled system downtime and modified by a secure MIRG administrator.

2. Flow of Events

2.1 Basic Flow

- Remote CDSS processing is temporarily disabled during scheduled system downtime.
- Administrator enters new schema information (which corresponds to a particular HIS and desired output parameter) or updates an existing schema (e.g. creates a new schema version).

- Schema is replaced in MIRG's repository and assigned a unique version name.
- CDSS is re-enabled.

3. Preconditions

3.1 Use case can only be invoked by an authenticated user with the appropriate privileges (e.g. MIRG Administrator).

4. Postconditions

4.1 New schema version is added to repository and will be referenced in future CDSS requests as applicable.

5.1.2.2.3 Publish Schemas in Registry/Repository

Use case package: MIRG Interoperability Support (figure 5-C) Use case number: 3

1. Brief Description

MIRG Administrator can register CDSS schemas with medical domain repositories for discovery within the medical community. These repositories are generally publicly accessible, although certain registration requirements may be enforced. *This is an excellent way of making schemas publicly available, without requiring that sufficient security be implemented at the MIRG server to make schemas publicly available to external non-registered users (e.g. guarding against schema integrity attacks).*

2. Flow of Events

2.1 Basic Flow

- MIRG Administrator registers with the repository to publish metadata.
- Most current version of schemas is updated as required.
- Repository service manages metadata for public access.

3. Preconditions

3.1 Schema is validated.

4. Postconditions

4.1 Schema is available for discovery within medical community.

5.1.2.2.4 System Maintenance

Use case package: MIRG Interoperability Support (figure 5-C) Use case number: 4

1. Brief Description

MIRG administrator must provide most current version of CDSS tools, corresponding transforms, and new medical databases. Concept of resource management applies since system users cannot be requesting CDSS processing when the version is being updated.

2. Flow of Events

2.1 Basic Flow

- Remote CDSS processing is temporarily disabled (also possible that this is handled by a concurrent versioning system as will be discussed in the design section).
- Administrator updates system with most current version of tools or data to be visible to other MIRG users.
- CDSS processing is re-enabled.

3. Preconditions

3.1 Use case can only be invoked by an authenticated user with the appropriate privileges (e.g. MIRG Administrator).

4. Postconditions

4.1 Most current version of medical data and CDSS tools is available.

5.1.2.2.5 Secure Login

Use case package: MIRG Data Interoperability (figure 5-C) Use case number: 5

1. Brief Description

A secure login is required prior to allowing MIRG Administrator and Internal MIRG Users to update schemas stored in MIRG's Medical Schema Repository. This repository can be thought of as the secure research repository, as opposed to the registry/repository services referred to in **Publish Schemas in Registered Repositories** where schemas are made publicly available to non-registered MIRG users.

2. Flow of Events

2.1 Basic Flow

- User enters security information.
- Application logic verifies user security and access parameters stored at the server.

3. Preconditions

3.1 Pre-set list of user security levels have been established for system Administrators and high security Internal MIRG Users.

4. Postconditions

4.1 User has been securely logged on and may update system resources.

5.1.2.2.6 Access Standardized Metadata via MIRG Repository

Use case package: MIRG Data Interoperability (figure 5-C) Use case number: 6

1. Brief Description

Registered MIRG users have read-only access to metadata stored in MIRG's repository. Note that this does not preclude external non-registered Internet users from viewing this data through registry/repository services.

2. Flow of Events

2.1 Basic Flow

- User specifies desired schema through a graphical user interface.
- MIRG repository manager retrieves specified schema.

3. Preconditions

3.1 User is registered (e.g. **Authenticate User and Establish Access Parameters** is successful).

4. Postconditions

4.1 Schema is displayed to registered user.

5.1.2.2.7 Standardize Medical Data

Use case package: MIRG Data Interoperability (figure 5-C) Use case number: 7

1. Brief Description

Internal MIRG users must standardize medical data based on **Invoke Knowledge Sources to Access Ontologies** (for access to external system resources) use case. This is a simple use case and is currently performed manually.

2. Flow of Events

2.1 Basic Flow

- User standardizes medical databases based on external authorized vocabularies.
- If the database includes missing values, **Standardize Medical Data** is extended by **Replace Missing Values** (discussed in the **Validate Data** use case in section 5.1.2.1.6).
- Using standardized data XML-based medical schemas are generated (this is currently performed manually).

3. Preconditions

3.1 Non-standardized medical data is available.

4. Postconditions

4.1 Standardized medical data is available for CDSS 'training' within MIRG.

4.2 Standardized metadata (represented as XML schemas) is available for use by **Publish Schemas in Registry/Repository** and **Update MIRG Medical Schema Repository** use cases.

5. Extension Points

Manual Standardization, Replace Missing Values

5.1.2.2.8 Manual Standardization

Use case package: MIRG Data Interoperability (figure 5-C) Use case number: 8

1. Brief Description

Standardize Medical Databases is an abstract use case of which **Manual Standardization** is a concrete instantiation. This use case inherits all the functionality of the parent and is created to distinguish between potential future **Automatic Standardization** use cases.

2. Flow of Events

2.1 Basic Flow

- As in **Standardize Medical Data**.

3. Preconditions

3.1 Non-standardized medical data is accessible.

4. Postconditions

4.1 Standardized medical data is available for CDSS 'training' within the MIRG.

5.1.2.3 NICU CDSS Support Package: Use Case Diagram and Descriptions

Use cases in the NICU CDSS Support package relate specifically to the NICU client and are included because they detail the specific environment with which MIRG's 'system' must interact.

Figure 5-D NICU CDSS Support

5.1.2.3.1 Prepare NICU Repository Data for CDSS Processing

Use case package: MIRG CDSS Support (figure 5-D) Use case number: 1

1. Brief Description

Prior to transmitting repository data to MIRG's CDSS, the NICU client must convert the data into standardized XML format.

2. Flow of Events

2.1 Basic Flow

- NICU repository logic retrieves data to be transmitted for processing (specified by patient identifier, date and time). Note that it is possible for future transformations to remove this patient identifier to ensure patient confidentiality during meta-analysis.
- NICU client application logic automatically retrieves corresponding MIRG schema by invoking **Access Standardized Metadata via Repository**.
- Client application logic performs automatic transformation of data into XML format for transmission (based on standardized ontologies described by MIRG schemas).

3. Preconditions

- 3.1 Successful secure login to access NICU repository.
- 3.2 Patient data exists and is present in NICU repository.

4. Postconditions

- 4.1 Data is ready for processing by MIRG CDSSs.

5. Extensions

Access Standardized Metadata via Repository

5.1.2.3.2 Device Scheduling (Real-time/interrupt driven and polled)

Use case package: MIRG CDSS Support (figure 5-D) Use case number: 2

1. Brief Description

NICU must update the NICU data repository with sensory medical device data (oxygen monitors, pulse-oximeters, cardiac monitors, etc.) on a periodic or interrupt driven basis. It is possible that the NICU Administrator may assign device priority levels to manage the order in which real-time device repository updates are scheduled. For polled devices the Administrator must also establish the periodic refresh rate. Additionally, it is suggested that a device that normally updates periodically will generate an interrupt if data deviates from an acceptable threshold range. All functionality described in this use case is constrained by the results of a feasibility analysis once the NICU repository is implemented. The implementation details of this use case are left as future NICU-based work.

2. Flow of Events

2.1 Basic Flow

- Relevant medical device updates are suspended.
- NICU Administrator sets priority, periodic update and/or interrupt level for each device.
- NICU administrator sets device specific threshold.
- This data is used by **Real-time Update of NICU Medical Data Repository**.
- Device updates re-enabled.

3. Preconditions

3.1 Successful secure login to access NICU medical device interfaces (e.g. by NICU administrator).

3.2 Device interfaces established.

4. Postconditions

4.1 Device schedules set for periodic polled and/or interrupt driven updates.

5.1.2.3.3 Real-time or Polled update of NICU Medical Data Repository

Use case package: MIRG CDSS Support (figure 5-D) Use case number: 3

1. Brief Description

Based on **Device Scheduling** parameters, the NICU's medical data repository is updated.

2. Flow of Events

2.1 Basic Flow

- Device interrupt is generated or device timer is triggered (in case of polled devices).
- Device sends data to repository for update.
- Data is added to the repository and referenced by patient ID and time.

3. Preconditions

3.1 Device parameters are set.

4. Postconditions

4.1 Repository is updated.

5.1.3 Non-functional System Requirements

In addition to the previously described use cases, the system architecture is also influenced by factors such as: distribution, usability, reliability, performance, design

constraints, security requirements and interfaces. These factors will augment the functional use cases when creating the analysis and design classes.

5.1.3.1 System Distribution (Network Topology)

- Internet connectivity is employed for establishing secure connections with external registered users.
- Clinical implementation will be based on a Hospital Intranet - this implies that the MIRC server is installed at the Hospital side (standardization issues are still important since system can be accessed by other HIS units).
- Non-registered Internet users may view MIRC schemas via public registry/repository structure in the medical domain (e.g. not maintained by MIRC but useful in promoting sharing of standardized CDSS metadata).
- Network topology will be discussed further in the deployment model (see section 5.3.1).

5.1.3.2 Usability

- The NICU side (or any HIS client) requires a highly intuitive and user-friendly interface. Creating the human computer interface at the physician side is left as future work and will be implemented by a Human Computer Interaction specialist. The required training time for NICU staff to become proficient at using this system must be very short.
- External registered Internet users require a much simpler, non-customised user interface that can be accessed through a browser configuration.
- Power users (MIRC researchers) will attend a short seminar on XML and the benefits of standardization and metadata publication.

5.1.3.3 Reliability

- System availability must be high since the NICU and other HISs operate around the clock. An availability of greater than 95% is required. System maintenance, such as schema and tool upgrades, is rare and will occur at night and with prior notification.
- Scheduled downtime is only a problem if zero downtime is a system requirement. This could potentially be handled in the future by using a versioning approach such as that provided by an open source Concurrent Versioning System (CVS). A CVS manages changes to documents stored in a shared repository structure. If the system must address zero scheduled downtime, updating the schemas would

involve creating a new schema with a new version number and the application logic (employing CVS functionality) would look for the newest version - the same versioning logic is required for new versions of CDSS tools. Implementing concurrent versioning functionality is left as future work.

- The Mean Time Between Failures (MTBF) must be low. There is no real reason for failure on the part of the CDSS; MIRG's systems are reliable and have been tested over many years. The main cause of failure is likely to be through connectivity and environmental (operating system, processor) failures. For a low MTBF framework implementation should target a highly robust (and low cost) Linux Server environment.
- The system must provide extremely high accuracy in terms of both the results produced by the CDSSs and the transmission of data.

5.1.3.4 Performance

- The average response time for a transaction request from the NICU must be as short as possible. NICU users will be assigned a higher processing priority than non-NICU users.
- The response time for an NICU transaction should be under 2 minutes. At this point it is difficult to estimate an exact figure because MIRG has only worked with training CDSSs based on entire patient datasets - the processing of which can take hours. Using the "trained tools" to predict the outcome for one patient has not yet been implemented.
- The system's capacity should ideally be able to handle all requests from HIS users and external registered users. To this end the implementation will target a robust scalable server environment (such as Linux and MySQL) where appropriate resources (e.g. processors) may be added as necessary.

5.1.3.5 Design Constraints

- The framework must adhere to medical standards specified by the UMLS or other standardized terminology servers. UMLS is an open-source server of which MIRG is a registered user.
- XML is chosen as the standard format for data representation.
- Coding is based on Java and applicable libraries.
- Initially data access shall be accomplished through open source DBMS solutions such as MySQL using Java DataBase Connectivity (JDBC).
- Must work with MIRG CDSSs.

5.1.3.6 Security

- The Internet will be used for external connections; this could be accomplished on a MIRC network if a Wide Area Network (WAN) was installed. In an insecure network like the Internet two security options are possible: (1) security and confidentiality requirements can be achieved through encrypted tunnels such as Virtual Private Networks or (2) by employing application level encryption such as that offered in conjunction with Public Key Infrastructure (PKI) services.
- It is worthwhile to note that some PKI Certification Authority (CA) Software is becoming available as Open Source. In addition, Health Canada has their own PKI CA to support health care research initiatives such as the framework proposed in this work.
- Security issues will be addressed further in the deployment model.

5.1.3.7 Interfaces

User Interfaces

- Physician-based interface at the NICU.
- Web interfaces for MIRC web-based client's and MIRC administrator - this interface must include facilities for online access to MIRC's medical schemas.

Hardware Interfaces

- Hardware interfaces are required to communicate with NICU devices for device scheduling and for polled and real-time updates.
- Interfaces with wireless NICU devices for generating remote alerts.

Software Interfaces

- Interfaces are required to communicate with MIRC's CDSSs. XML will be used to convert the data into an interoperable format that can be read by CDSSs. Currently MIRC's ANN is used in a research environment where functionality is invoked through the Matlab command prompt. The CBR provides a specialised graphical user interface for NICU client. Neither tool currently provides interfaces to allow invocation from application programs.

5.2 Analysis Model

The IEEE definition of an architecture is “*The highest level concept of a system in its environment*” [IEEE 1999]. Based on this definition, the architecture is incrementally adapted to match the implementation environment. The purpose of this section is to transform the functional requirements described by the requirement model’s use cases into a candidate system architecture. The analysis model represents an abstract, conceptual view of the architecture; classes are defined generally and are often abstractions of several classes and subsystems that will be developed during design. Classes in the analysis model focus on handling functional requirements, non-functional requirements are left for the design and implementation models. Therefore, analysis classes are not described by their operations but by their roles, responsibilities and high-level attributes.

5.2.1 Class Design and Packaging Strategy

This section describes the analysis model’s packages and the significant classes associated with each package; packaging and class identification strategies are discussed. Classes have been allocated to three high level packages named: Application, Persistence and Boundary. Typically, analysis classes are categorised (stereotyped) as either control, entity or boundary classes (see Appendix A for examples of these stereotypes and a brief overview of UML packages).

5.2.1.1 Boundary Classes

Boundary classes model interactions between the system's surroundings (represented by the actors) and itself; as a general analysis rule there is one boundary class for each use case/actor pair [Jacobson 1999]. Boundary classes are classified as user

interface classes, device interface classes or system interface classes. Because this system is web-based there are numerous user interfaces responsible for co-ordinating actor/system interactions (these are emphasised with a note in figure 5-E). Additionally, NICU devices and sensors that generate external events necessitate device interfaces.

There are two main packaging strategies for boundary classes: either isolate the boundary classes from the rest of the model, or, place the boundary classes with the entity and control classes with which they are functionally related. Because the system interface is still being developed and will undergo further changes, particularly from the NICU perspective, the decision was made to encapsulate interface functionality so that changes would only affect the interface boundary package. This relates to the principle of promoting flexibility and scalability of medical system components. One exception is the interface between 'Internal MIRC User' and the 'UMLS Knowledge Source Server' control class; because this interface is stable it is positioned within the same package as the entity and control class with which it interacts (e.g. the Application package).

The Boundary package's classes and the actors with which they interact are shown in the class diagram in figure 5-E. Note that a class such as '*Device Interface*' is an abstract boundary class from which an instance will be created for each instance of the '*NICU Medical Device*' that requires communication. The same applies to '*CDSS System Interface*'; an instance of this abstract class is created for each CDSS that is added to the system.

Figure 5-E Boundary Classes

5.2.1.2 Entity Classes

Entity classes model key system concepts that are persistent and stored over the life of the system; these classes are normally mapped to a database in design. The architecture's main entity classes are allocated to the Persistence Package. However,

certain entity classes have been packaged with their controlling class in circumstances where the entity class does not represent so much a *system concept* as a container for storing information. An example is given by 'Audit Log', which is grouped in the MIRC Server package and where specific CDSS transaction information is stored, not necessarily for the entire life of the system. Additionally, entity classes considered external to the core framework functionality, such as the 'RealTimeDevicePriorities' and 'DevicePollingSchedule' that are required by the NICU client are stored locally to the NICU (e.g. within the same package as its functionally related boundary and control classes) for ease of distribution.

The Persistence Package (see figure 5-F) includes storage for all XML-based medical schemas and transforms ('MIRC Medical Schema' and 'MIRC CDSS Transform'). The class diagram shows an inheritance relationship between these XML-based artefacts and the W3C standards (or recommendations) on which they are based - for example 'MIRC Medical Schema' inherits from 'XML Schema 1.0'.

The Persistence Package also contains client registration information required by the User Registration, and Authenticate User and Establish Access Parameters use cases. Along with the XML-based 'MIRC Patient Data' these three entity classes require the highest level of security and may only be accessed by the MIRC Administrator; it is recommended that they be isolated in the design model into a high security package to protect sensitive patient data and access privileges to this data.

Figure 5-F Persistence Classes (Persistence Package)

5.2.1.3 Control Classes

Control classes model use-case specific control behaviour, controlling and co-ordinating system objects that implement the use case functionality; a control class is often related to a particular use case. Within this system architecture control classes have been assigned to the Application package, which contains three sub-packages: MIRG Server, NICU Client and UMLS Server, which models classes that represent the external Unified Medical Language System with which MIRG actors must interact to achieve data standardization. Classes belonging to NICU Client and MIRG Server are positioned in

the same package because their classes are functionally related; functional relativity is identified between classes exhibiting extensive intercommunication and where behaviour changes to one class have an effect on the other's state.

5.2.1.3.1 MIRC Server

The MIRC Server sub-package contains all control classes required to manage diverse CDSS transactions, co-ordinate MIRC resources, perform user authentication and handle errors. Figure 5-G presents a collaboration diagram showing the main classes contained within the MIRC Server sub-package and their functional interactions. Collaboration diagrams are often used in analysis because the focus is on identifying object responsibilities rather than showing detailed chronological sequences (as will be presented in the design model). The collaboration diagram emphasises interactions at the MIRC Server and NICU Client level, since these are the most architecturally complex packages containing system control behaviour.

The 'CDSS Request Transaction Manager' control class manages requests from NICU, Internal MIRC users and external (Internet-based) registered users as well as controlling other 'worker' classes such as 'Data Transformer' and 'Validate Data'. This class also directly accesses two entity classes: 'Alert Thresholds' and 'Audit Log'. All transaction data is stored in the audit log for system evaluation purposes and for retransmission in case of error, the 'Error and Recovery Manager' control class handles the later functionality. 'Error and Recovery Manager' was created to manage processing and transmission errors that may occur during the **Invoke CDSS Processing** use case. This is based on the strategy that it is best to encapsulate the main flow of events and alternate or exceptional flows in different control classes in order to increase long-term system maintainability.

The 'XML Resource Manager' control class performs resource management of XML resources, such as the schemas and transforms required for CDSS processing. The idea here is to employ a control class to de-couple boundary and entity classes in order to increase the system's tolerance to changes in the system boundary. This also removes use-case specific behaviour from the resources, making them reusable by different use cases. Employing such a resource manager means that schemas can be invoked externally and therefore need only be stored at the server side, thus minimising data duplication and providing a defined source for medical metadata.

'Data Transformer' and 'Validate Data' are stereotyped as <<business logic>> classes and are included in this package following the idea that classes with heavy intercommunication should be grouped together. Both classes have intensive communication with 'CDSS Request Transaction Manager' and will change state (from idle to processing) based on interactions with this control class.

The class diagram presented in figure 5-H shows all interactions between control, entity and boundary classes required to perform client validation and manage access to system resources. The 'Client Validation' control class authenticates users based on accessing the appropriate entity classes. 'Schema Resource Manager' and 'CDSS Version Controller' ensure respectively that the most current version of the medical schemas and tools are available for CDSS processing.

Figure 5-H Registration and Resource Management Class Diagram

5.2.1.3.2 NICU Client

The NICU client provides generic functionality required to implement this framework in hospital information systems. As mentioned in section 2.1.3, the development of the NICU online data repository is only beginning at the time of completion of this work. Therefore, the interaction with the NICU client must be as extensible and flexible as possible to accommodate certain changes in the NICU system requirements that will arise during implementation. The prototype implementation presented in section 5.4 will not cover the NICU client-side; integrating with the NICU

data repository is left for future work. An outline of the NICU classes is given in figure 5-I; this is subject to change based on the repository's technical implementation details. The 'NICU Controller' class, in conjunction with the 'NICU CDSS Boundary', is responsible for managing client CDSS processing requests and will be modeled in more detail in the upcoming design section.

Figure 5-I NICU Client Class Diagram

5.2.1.3.3 UMLS Server

This package is included for clarification purposes because the system (represented by a request from 'Internal MIRG User') must interact with the 'UMLS Knowledge Source Server' for data standardization purposes. This interaction is shown in the class diagram presented in figure 5-J. This work will not modify current UMLS interfaces, however, as mentioned in section 3.2.2, future UMLS work will involve modifying the UMLS Knowledge Source Server to accept XML requests. This presents the potential to automate the medical data standardization process as future MIRG work.

Figure 5-J UMLS Classes

5.3 Design Model

The analysis model defined abstract classes where the implementation details were ignored. The framework's design model extends the analysis model to define more 'physical' classes and subsystems. While detailed implementation components (e.g. binaries and executables) are not shown, the systems and subsystems required by the implementation components are. In design, analysis classes are developed, and detailed

layering dependencies between systems and subsystems are shown. Jacobson describes the abstractions in the design model as “*a blueprint for the implementation and a straightforward mapping to implementation*” [Jacobson *et al.* 1999]. In this way the design is created using the analysis as a primary input and expanding on the generalized analysis classes to address the selected implementation environment.

However, before refining the analysis classes into detailed design classes, design issues that were glossed over in the analysis need to be examined in greater detail. First, the suggested physical deployment is shown and technological trade-offs in terms of distributed technologies and security issues are discussed. This is followed by a discussion of architectural considerations, such as distribution patterns and assigning functionality to deployment nodes. This section concludes with the presentation of design artefacts, specifically detailed class and subsystem diagrams. The goal at this point is to have developed a design such that the implementation can be decomposed into subsections for future work.

5.3.1 Deployment Model

While the deployment model is often left to the end of the design model, if deployment issues are known they can be presented earlier, as network topology significantly impacts other architectural choices. The deployment model describes the physical distribution of the system across a set of processing nodes; a node represents either a processor or a device. The deployment model is useful in terms of giving a high-level physical representation of the system; Jacobson states “*physical network configuration has a significant impact on architecture, in terms of active classes and the distribution of functionality across the network*” [Jacobson *et al.* 1999]. In addition to

5.3.1.1 Justification for Choice of Distribution Technology

In this thesis XML has been chosen as the means of achieving standardized messaging between distributed applications. Section 2.2 (2.2.6 in particular) rationalised this choice by discussing distribution situations where it is appropriate to employ XML. However, this work would not be complete without a discussion of alternative distribution technologies. Many consider XML the future 'distributed application protocol' through which information will be exchanged in environments that do not share common platforms but desire data interoperability. Traditionally this interoperability has been achieved through Distributed Object Technologies (DOTs) such as the Object Management Group's Common Object Request Broker Architecture (CORBA).

CORBA is an open standard developed by the Object Management Group that provides communication between objects on heterogeneous platforms [Gomaa 2000]. Based on the Object Request Broker middleware, CORBA server objects provide services that can be invoked by the client(s). The server object interface is defined based on an Interface Definition Language (IDL). Additionally, CORBA provides essential features in distributed computing, such as data access control, security, error handling and robust interoperability.

Watson explains the difference between the Interface Definition Languages used in distributed object technologies and markup languages such as XML. *"In contrast to markup, which is used for the long-term storage of human-readable data, IDLs define the smaller packets of transient, machine readable data that are exchanged between the components of a distributed application when some particular event occurs"* [Watson 2001]. It is important to realise that XML and DOTs, such as CORBA, are not mutually exclusive. The Object Management Group is currently working to integrate XML into the

CORBA technical infrastructure, so that new XML-based applications will ‘plug and play’ with current CORBA-based applications [OMG 1999].

In this thesis’ framework, where both applications and humans must read data, and medical data standardization efforts require XML’s data representation flexibility, it is postulated that XML is better suited than CORBA. The IEEE report comparing XML and DOTs stated that many respondents were not using CORBA because it was too inflexible: “*CORBA’s type system hurts more than it helps in highly evolutionary environments*” [IEEE 2002]. Health care is a highly evolutionary environment with few existing public interfaces. The IEEE report concluded that XML works better for messaging in loosely coupled systems with high messaging requirements and CORBA in tightly coupled ones. The architecture in this thesis is based on loosely coupled interactions, where applications interoperate as loosely-coupled system using Internet protocols and based largely on asynchronous messaging.

While the Object Management Group developed CORBA as an open-source initiative, many CORBA-based application products have licensing requirements and associated fees. CORBA is also generally regarded as having a high degree of programming complexity. In the Medical IDEAs Research Group, where students will meet the majority of implementation requirements, reducing the complexity of application integration through the use of XML-based technologies is advantageous. It can be concluded that for meeting the stated objective of achieving an interoperable and integrated Internet-based CDSS framework, open-source flexible technologies such as XML present the best solution. Smith corroborates this point, claiming, “*XML can be*

implemented more quickly and easily and at a fraction of the cost of traditional [DOT] approaches” [Smith 2001].

However, if this framework’s architecture was extended to a more tightly coupled environment, where the majority of the data was meant for processing and storage with no need of human intervention, CORBA could effectively be integrated into the communication model and XML data could be transmitted over CORBA. This was seen in the literature review where several architectures transmitted XML documents over a CORBA architecture to achieve interoperability between diverse units of the hospital information system [Crichton *et al.* 2001, Muller *et al.* 2001]. If a CORBA-based architecture suited MIRG’s goals, the current framework could be extended to include this functionality.

5.3.1.2 XML Messaging-based Distributed Computing

The deployment diagram indicates that the Simple Object Access Protocol (SOAP) or XML Remote Procedure Call (XML-RPC) are both considered as future means of transmitting XML data between distributed network locations. These represent only two transmission possibilities; Internet communities are repeatedly proposing the use of XML in network protocols and distributed applications- XML-RPC, SOAP and XML Metadata Interchange (XMI) are only a few examples [Bonifati *et al.* 2001].

Briefly, XML-RPC provides a specification and functionality to allow distributed software running on different operating systems and in different environments to make procedure calls over the Internet. HTTP is used as the transport mechanism and XML for the syntactical encoding. The idea behind XML-RPC is to provide “*Simple cross-platform distributed computing, based on the standards of the Internet*” [XML-RPC 2001].

The Simple Object Access Protocol provides messaging for web services applications and is the standardized enveloping mechanism for communicating messages and remote procedure calls using XML. A web service is an interface that describes a collection of operations that are network-accessible through standardized XML-messaging [WSCA 2001]. Exploring the potential of web services in health care delivery would be interesting future work, offering a strong case for employing standardized SOAP envelopes in XML messaging. The W3C describes SOAP as: “*A lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses*” [W3C SOAP 2000]. While SOAP is employed at the XML messaging layer, simple HTTP is used for communication at the network layer.

Employing protocols such as SOAP is beneficial because the envelope level can manage issues such as guaranteed delivery, transmission errors, and routing information [Gregory 2001]. However, examining the benefits and differences between various XML messaging protocols will be left for future system implementation; for now the important part is that these protocols exist, are evolving rapidly and will provide appropriate error recovery and security to allow successful implementation of Internet-based transmission of medical data.

5.3.1.3 Data Security

Security requirements are paramount in Internet-based architectures charged with the delivery of highly sensitive patient data, such as that described in this thesis. Data security must be considered at three levels: the security of medical data, Internet-based

security concerns, and new security issues raised by the introduction of XML-based metadata.

While the Internet offers many advantages to the health care environment in terms of increased interoperability for providing improved patient support, increasing the availability of information, and reducing costs, Gritzalis observes that “*many [in health care] are reluctant to use the Internet due to the security risks entailed*” [Gritzalis *et al.* 2001]. Therefore, in order to reap the advantages of Internet-based health care frameworks, security requirements need to be defined. Specifically, to ensure secure web-based data transmission one requires data confidentiality (protecting the patient identifier), data integrity (ensuring the data is not altered during transmission), authentication (to validate the document’s sender) and non-repudiation (the sender cannot deny either sending the document or its contents). Tidwell states that the Secure Sockets Layer (SSL) provides the first three functions and claims that the XML Security Suite can provide the fourth [Tidwell 2000]. The SSL manages the security of network transmissions through the use of session key encryption nested within public key encryption based on open Public Key Infrastructure (PKI) standards. Employing PKI implies the need for a PKI Certification Authority (CA) for positive authentication - SSL without such an X.509 certificate does not handle positive user authentication (e.g. authentication is provided for the life of the session only with no knowledge of who the user actually is, such as would be obtained through a certification authority).

This design will largely avoid making final security decisions because the XML Security Suite standards are still under development and the XML-based structure of this framework makes employing these standards a logical choice. Currently the W3C is

developing encryption mechanisms for secure data transmission over lightweight protocols such as HTTP. The mission of this Working Group is to “*develop a process for encrypting/decrypting digital content (including XML documents and selected portions thereof) and an XML syntax used to represent the (1) encrypted content and (2) information that enables an intended recipient to decrypt it*” [W3C Encr. 2002]. Further XML-based security initiatives address the concept of XML digital signatures.

In general, XML helps to alleviate security issues because, by using schemas to separate patient data from the metadata, the schemas can be made public without risking the integrity of the patient data. At no time will this framework make patient data publicly available and a patient identifier that is significant only to the invoking client will represent the patient data transmitted for processing. A security concern related to the Electronic Patient Record that is also relevant in this system is to “*ensure that every data output is correctly related to the corresponding patient*” [Wolff *et al.* 2001]. If the patient identifier in the XML document were corrupted the results would be meaningless. Additionally, data integrity during transmission is of particular importance. Wang describes an integrity attack to the transmitted patient data whereby the attacker may potentially intercept the data and replace it with any arbitrary content, resulting in the diagnostic report being altered and producing erroneous treatment plans [Wang 1999]. This would be disastrous in the case of clinical decision support systems, because if the physicians cannot rely on the integrity of the results, the system becomes useless and potentially dangerous. Ensuring the integrity of medical data during transit is left as future work and is a

necessity before the framework's 'interoperability' can be viably offered to medical institutions.

One of the stated aims of this system is to promote standardized interoperability by allowing external and internal users to view MIRC's medical schemas via the Internet. Because the XML schemas are meant to be publicly accessible, the main concern is that unauthorised changes not be made to the metadata as it forms the basis of CDSS processing in this framework. It is critical that the framework ensures metadata security through data repository security mechanisms. Megginson corroborates this point, stating that XML raises new security concerns due to its reliance on external shared resources [Megginson 2000]. In the initial framework implementation, it is recommended that remote modifications to schemas and tools not be possible and all changes to data and CDSS tools be accomplished locally by the System Administrator.

5.3.2 Architectural Considerations (Distribution Patterns)

While it is clear from the deployment model (figure 5-K) that the framework should take advantage of Internet connectivity, specific distribution issues have not yet been discussed. Most significantly, issues such as defining the framework's architectural patterns must be addressed before functionality can be allocated to either the MIRC server or the clients.

The distribution pattern of the architecture described in this thesis follows a web-based three-tiered client-server architecture. As can be seen from the deployment model, different processor nodes represent both clients and servers. The three-tier architecture extends the client-server architecture by dividing system functionality between three logical nodes: application services (the server), business services (the client) and data services (the database server). Additionally, because the architecture is web-based, the

traditional three-tier architecture pattern must also incorporate web architecture patterns, specifically the issue of whether a thick or thin client architecture is employed must be addressed.

In addition to reaching an extremely broad client base (made broader through medical data standardization efforts), one of the main benefits of a web-based architecture is to provide the scalability to make changes to the application software without requiring that it be re-deployed at the client(s). To this effect, the client should be kept as 'thin' as possible with most of the application and business functionality being executed on the server. This configuration is clearly beneficial in the deployment of research-based CDSS software in a distributed HIS with potentially numerous clients - if every upgrade to the CDSS tools required a parallel upgrade to the client the scalability of the system would be severely diminished.

Because most of the application and business logic in a thin client architecture is executed at the server side the client only requires minimal configuration, such as a standard web browser, making the architecture deployable across a wide variety of client configurations. External clients of MIRG's CDSSs are patterned on a thin web client architecture. Conversely, in a thick web client architecture the client can perform significant processing via mechanisms such as Java applets - at the risk of losing application portability to a broad client base.

The decision must be made as to whether the NICU client (which is representative of generic HIS units) should be patterned as a thick or thin client, keeping in mind the potential for expanding MIRG's clinical decision support to other HIS units, as well as MIRG's desire to provide clinical decision support to locales external to the HIS.

The decision was made to model the NICU processor (node) as a thick-client architecture. One of the motivating factors behind the choice of a thick client architecture is that a customized user interface is required at the NICU client side. The user interface (UI) required by the NICU must be highly user-friendly and developed and refined based on human factors studies. The user interface should also assist the user in navigating the site and be as simple as possible, allowing the user to easily transmit retrieved patient data for processing based on minimal user preferences. Additionally, the choice of a thick client NICU architecture is reasonable, based on the fact that within a hospital information system environment a certain client configuration can be assumed. Nevertheless, for maximum scalability and maintainability the framework will minimise client processing to include only essential user interface functionality.

5.3.2.1 Client-side Considerations

There must be a clear delineation between local client-side work, such as managing the NICU real-time repository and controlling medical devices (the design of which is beyond the scope of this thesis), and the work of the client's web-browser. It would not be practical for physicians and HIS staff to perform manual data entry of XML-based patient data for CDSS processing. Rather the NICU User Interface must provide the functionality to automatically access the NICU repository data, retrieve patient information based on a specific time and patient identifier, and automatically transform this data into XML for transmission.

In an initial clinical implementation, NICU repository data will not be originally stored in XML format, and the most likely candidate would be a relational model. As stated in section 3.3.4, to ensure maximum interoperability and to reduce a proliferation of gateway interfaces, it is the client's responsibility to transform data into the

standardized format. The client can perform this relational transformation to XML via the Java APIs for XML (JAXP) programming based on the Document Object Model (DOM) described in section 2.2.4. Relational conversion to XML is a common problem and will be discussed and implemented in section 5.4; additionally a standardized medical framework greatly simplifies this conversion problem as the relational database's column values used in the NICU will be based on the same standardized medical ontologies as the XML schema tags. Employing emerging XML-centric data management techniques for directly transforming SQL data types to/from XML data types, such as those based on the ISO 9075-part 14, the SQL/XML specification, will be definite future work as implementations of these transformations become available.

5.3.2.2 Data Distribution

Data is distributed in this system between the MIRC server and the NICU, in that patient data collected at the NICU client is stored at that location. As described, NICU data must be automatically transformed into an XML-enabled format, which requires access to the applicable XML schemas. Therefore the decision must be made where to store XML artefacts required for CDSS processing, specifically whether to employ central metadata storage or data replication. Clearly, one of the drawbacks of central metadata storage is that the server can become a bottleneck to processing. However in this system where schemas will be updated frequently and a defined authoritative source for metadata is required, it appears reasonable to choose central metadata storage and to address bottleneck issues by supporting high-volume server throughput.

In this design, interoperable clinical decision support relies on stored XML Schemas and Stylesheets, that respectively validate and transform data prior to processing, both of which are subject to change based on corresponding changes to the

CDSS tools. Changes can be based on improvements developed through ANN training, for example discovering an optimum minimum data set for a particular outcome. As described, the schemas will promote broad applicability of MIRG's CDSSs by allowing external users to send interoperable data for processing. It follows then that the schemas must be stored at the server side. To minimise data duplication and increase flexibility, both the XML Schemas and the XSLTs will be stored at the server side only.

Validation of XML-enabled patient data occurs prior to data processing, and is based on XML schemas that combine rules containing acceptable NICU device values and acceptable CDSS inputs. As discussed in the use case analysis, it is also possible that this data validation be extended to replace missing data values. The decision whether to perform data validation at the client or server side must be considered. At first it appears reasonable to validate data at the client prior to transmission to eliminate the cost of transmitting invalid data. However, because the NICU's translation to XML is automatic (as would be the case in any other integrated HIS unit), the probability of receiving invalid data is low. Additionally the cost of transmitting invalid data is insignificant in comparison with the loss in flexibility and software distribution issues associated with client-side data validation, which would require that the XML schemas be stored locally and that the validating code be performed using distributed Java applets. It follows that transforming the data for CDSS processing is also performed at the MIRG server conditional on successful data validation.

Data distribution issues are also relevant in generating CDSS-driven alerts. When a physician transmits data to a CDSS there are instances when the results will indicate extremely high probabilities of an adverse outcome. In situations that the physician

deems 'critical' he/she can specify, through client-side user interface parameters, that an alert be generated. Based on such an alert the XML-enabled response could be transmitted to physician's wireless devices using the Wireless Markup Language (WML) to alert physicians outside of the HIS unit of the problem. As an aside, XML's ability to interoperate with wireless devices, so essential in healthcare, is another indication of its suitability in this problem domain.

In this situation the physician decides the appropriate threshold for the CDSS output (e.g. if patient has a 60% probability of requiring greater than 8 hours of ventilation, generate an alert). However, if this computation were performed at the client side this would require that the client parse the XML-based CDSS result to determine the value of the <alert output> tag. As discussed, this is not ideal, since client-side processing should be kept to a minimum to ensure maximum scalability of the system to other HIS units. The proposed solution is to allow the physician to specify the alert threshold for each CDSS transaction through the customised UI, with the default being no alert generation. The alert threshold is then added to the automatically generated XML document, along with the other patient data and UI parameters. Subsequently, MIRG's server application receives and parses the XML document, storing the parsed CDSS-alert threshold value locally. The result obtained from CDSS processing is then compared with this threshold value, and the result is added to the final XML result that is transmitted to the NICU user and potentially other users through wireless devices. Local client side processing would handle alert transmission to the appropriate wireless devices.

5.3.2.3 Server-side CDSS Processing: Managing High-Volume Throughput

Server-side clinical decision support system processing is a CPU intensive task and as specified in the requirements, a short response time for CDSS processing is

required. It is possible that this low response time must be achieved in the presence of a high-volume of system requests. High-volume throughput can be achieved by applying additional CPUs, memory, etc., and by creating code that takes advantage of the multiple processors that are part of a symmetric multi-processing environment (SMP).

Symmetric multi-processing environments provide a computer architecture that improves system performance by making multiple CPUs available to complete individual processes simultaneously. Additional CPUs, operating concurrently and communicating via shared memory, can be added to improve performance and handle increased throughput. Operating systems and hardware agents are available to support SMP, and specific applications can benefit from SMP if the code allows multi-threading, such as is provided in a Java-based implementation (Java is a concurrent programming language).

An SMP-based architecture would typically be implemented by forking new processes, where a constructor function is used to create new instances of 'worker' classes (such as 'Validate Data', 'Transform Data', and 'Process Data by CDSS'). Worker classes can then be processed on different processors but under control of the same operating system and using shared memory.

A concurrent server system would be particularly advantageous in improving the throughput when performing the **CDSS Semantic Interoperability** use case (processing by more than one CDSS) for an individual client. In this instance, because the results from the diverse CDSS are not logically related it is possible that all processing could be executed concurrently. In the case of the **Hybrid CDSS Processing** use case, where CDSS output must proceed in a logical order, concurrent processing would not be beneficial and sequential server processing is required. However, in both instances

concurrent server systems could improve throughput by allowing one task to execute on the CPU while another is waiting for XML artifact retrieval from the MIRG repository.

5.3.3 Design Artefacts

Having identified the architecturally relevant design issues, the UML artefacts for describing the architecture will be presented- specifically, the design classes and the grouping of these classes into layers and subsystems. Design artefacts are often program language specific and user defined stereotypes may be employed (other than the pre-defined <<boundary>>, <<entity>> and <<control>> stereotypes used in analysis). This design model includes enabling packages, such as those offered by the Java Application Programming Interfaces (APIs), to facilitate the final mapping to the implementation model. While the top application layers are mainly traceable to the analysis model's Application package, the design model offers increased detail in terms of the business-domain layer, middleware layer and system software layer, and shows dependencies between the layers' subsystems.

Subsystems (and their contained classes) are assigned to one of five layers: the Interface (shown in orange), Application Specific (yellow), Business Domain (purple), Middleware (green) or System Software layer (pink). The Interface, Application Specific and Business Domains are viewed as the 'higher' layers, with the Interface and Application Specific layers containing application-specific (use case-specific) functionality and the Business Domain layer containing application general functionality. The remaining layers are considered 'lower' layers and contain generic, core functionality that is reusable by other systems. General layering guidelines state that the highest layers contain the most volatile subsystems. A volatile subsystem is one that is most liable to change when user requirements change, such as the Interface subsystem.

Lower levels are much less volatile and will only vary when the implementation platform changes. This layering overview is shown in figure 5-L; the same colour-coding scheme will be used throughout the design to aid in distinguishing between subsystems belonging to different layers.

Subsystems within each layer are designed to encapsulate behaviour and will ultimately provide implementation interfaces through which they may be referenced by external subsystems. It is common for classes that participate in the same use case to be assigned to the same subsystem (provided that the classes are not physically distributed such as those in the MIRG Application Server and the NICU Application Client subsystems). The goal of subsystem encapsulation is to allow system functionality to be decomposed along dimensions of concern, maximising system flexibility and reuse.

The class diagram in Figure 5-L shows explicit dependency relationships between the classes and subsystems of the higher layers with the lower level middleware and system software. In UML dependency relationships are depicted as dotted lines with arrows in the direction of the dependency. Dependency relationships provide rules for restricting the visibility between layers; the accepted modeling rule is that subsystems may only depend on subsystems in the same layer and in the next lower level. Adhering to the principle of low coupling between modules (subsystems and layers) and high coupling within modules reduces integration complexity at the implementation stage.

Figure 5-L System Layering and Package Overview

5.3.3.1 MIRG Application Server Subsystem

The subsystem is responsible for executing the majority of the use case functionality (the system's value-added software) described in section 5.1, and provides the application logic behind Web Server subsystem. Because the MIRG Application Server must accommodate different services, such as HIS transactions for NICU clinical decision support processing, data validation, external CDSS processing and schema resource management, this subsystem contains a services broker class. The 'MIRG Services Broker' class receives client events from the web server and determines the event type (e.g. Is this a CDSS request from a NICU user? A request to update a medical schema from an internal MIRG user? A request to view a schema from an external user?). Based on the determined event type the appropriate control class is invoked.

The motivation behind the 'MIRG Services Broker' is to accommodate potentially high-volume throughput situations where MIRG's web server is receiving requests from multiple HIS units and/or external users and parallel processing may be exploited to meet CDSS real-time requirements. Currently, priorities have been implemented at the 'MIRG Services Broker' level whereby processing HIS-based requests is allocated a higher priority than processing requests from external users. Because the 'MIRG Services Broker' receives all user requests, manages priorities and controls MIRG's clinical decision support systems (through provided CDSS interfaces), parallel CDSS processing could be implemented as required based on the need for greater response times or increased processing capacity. This would be accomplished using a symmetric multi-processing environment (section 5.3.2.3) in which the services broker would allocate processing to other 'worker classes' to accomplish CDSS processing, data validation, data transformation and/or resource retrieval.

When examining response time requirements it is worthwhile to take a step back and examine the context in which MIRG anticipates its clinical decision support will be used. From MIRG's viewpoint the important term to identify is 'support'; these systems are not meant to be all encompassing in terms of replacing the physician's logical decision-making process. With this in mind, clinical decision support should not be used in all patient situations, but rather in difficult circumstances where the physician requires additional information. By no means does MIRG advocate making 'snap' decisions based on CDSS output; rather this information is to be thoughtfully considered and evaluated. To this end, the time requirements for processing are more lax than one might originally suppose. While unacceptably long delays would obviously make the system unusable, instantaneous results are also not necessary.

Figure 5-M depicts the design level dependencies and detailed operations for classes in the MIRG Application Server subsystem. The 'MIRG CDSS Control' class coordinates classes required to perform CDSS processing. This class is responsible for initiating all XML-related processing and parsing required for clinical decision support, storing medical data for auditing purposes, and transforming MIRG's medical databases from a relational model to XML (discussed in section 5.4). 'MIRG CDSS Control' stores all session parameters (shown in the class' attribute list) for a particular request based on input received from parsing the received XML document. Using these session parameters 'MIRG CDSS Control' determines the appropriate schemas and transforms required for CDSS processing. For example, based on the desired clinical decision support output and the requested CDSS identifier, a specific XML schema will be retrieved by the 'XML Resource Manager'; similar logic applies when retrieving pre and post-processing XML

transforms. 'MIRG CDSS Control' also performs application logic for determining the sequence in which clinical decision support will be invoked in instances where multiple CDSS identifiers are present and hybrid processing is required.

The 'XML Resource Manager' controls all access to XML resources such as schemas, transforms and stored audit data. As noted in the requirement section, because the processing of data is based on accessing the stored schemas and transforms, any changes to the schemas must be made in 'batch-mode' during scheduled system downtime. It is highly recommended that the initial implementation be carried out in a test environment and migrate to a production environment only when testing is complete. This clear distinction between a test and a production environment is needed in order to protect the production data from destruction by testing.

Detailed chronological interactions between the MIRG Application Server classes are shown in figure 5-N's sequence diagram. This diagram explicitly shows all the interactions required for performing CDSS processing; these interactions form the basis of the technological validation described in section 5.4. Interaction diagrams required for performing secure remote CDSS processing and remote medical schema invocation (through MIRG's medical schema repository) are not included, while security and authentication issues have been discussed in this thesis their implementation and detailed design modeling are left as future work.

Figure 5-M MIRG Application Server Class Diagram

Figure 5-N MIRG Application Server Sequence Diagram

5.3.3.2 NICU Application Client Subsystem

The design classes at the NICU thick client side offer only marginal differences from the analysis model; the design level class diagram for the NICU Application Client subsystem is shown in figure 5-O. The relevant NICU classes are ‘Physician Interface’ (derived from analysis’ ‘NICU GUI Boundary’), ‘Transformation Control’ and ‘NICU CDSS Transaction Manager’. ‘Physician Interface’ provides all boundary class functionality required for the client to interact with the web server to achieve remote CDSS processing. The role of the ‘NICU Controller’ class identified in the analysis model has been divided between the ‘NICU CDSS Transaction Manager’ and ‘Transformation Control’ classes.

The ‘NICU CDSS Transaction Manager’ is responsible for managing client requests, interacting with the NICU Repository to retrieve patient data (via the ‘Repository Controller’ class), and determining (based on the relational database model) the appropriate schema required for converting the patient data into XML. This data is then forwarded to the ‘Transformation Control’ class that performs the conversion of relational patient repository data (selected by the physician through the user interface) to XML for transmission to the MIRG Application Server subsystem. Achieving this relational to XML mapping is discussed in detail in section 5.4. Developing a detailed NICU design is left for future work when more details and user requirements are known.

Figure 5-O NICU Application Client Classes

5.3.3.3 Business Domain Layer

The Business Domain layer contains classes that are more generic than those described in the Application layer, in that they can be reused by multiple application classes. The Business Domain layer comprises the following three packages.

1. **Business Entities:** Containing all the XML-related data (transforms, schemas and documents). Also contains generic support capabilities, such as parsing and processing.

Java APIs provide ready-made interfaces for accessing XML parsers and processors. Dependency relations between the business domain layer and the Java packages in the middleware layer were illustrated in the overview layer diagram presented at the beginning of section 5.3 (figure 5-L). Business entities are shown in figure 5-P.

Entity classes identified in the analysis model may be further stereotyped as either <<data abstraction>> or <<database wrapper>> classes. Examples of database wrapper class are given by the 'Medical Audit Log' and 'MIRG Medical Data' classes, which represent patient data stored in a relational database management system (RDBMS). Database wrapper classes hide details of the interface to the external RDBMS where the data is held.

All XML-based resources are currently stereotyped as persistence classes and are accessed by the MIRG Application Server subsystem's 'XML Resource Manager' class. The 'XML Resource Manager' class can be thought of as a data abstraction class, providing operations to read, write and access the data encapsulated in the XML-based persistence classes. A data abstraction class hides the details of how the internal data structure is represented. As XML-based DBMSs evolve, the possibility exists for incorporating <<XML database wrapper>> classes into the design model.

2. **External System Interfaces:** Containing the interfaces to MIRG's CDSSs and that to the UMLS Knowledge Sources.
3. **Security:** Containing all classes to provide transaction security and registration (this is left for future implementation).

Figure 5-P Business Entities

5.3.3.4 Middleware

The middleware layer contains subsystems for platform independent services such as those offered by the Java and org packages. The org subsystem contains all packages required to support XML data manipulations, such as the Xerces parser and Xalan processor. The java.sql and jdbc package provide all classes required for connecting to an SQL-based relational database management system (shown in the system software layer).

The web browser and web server subsystems are also placed in this layer and make use of java.net for connectivity, java.applet for client side processing and java.awt for client side user interfaces. Both the browser and server depend on HTTP and SOAP subsystems found in the system software layer. The middleware dependencies with the upper layers and with the lower system software layer are shown in figure 5-L's overview layer diagram.

5.4 Technology Validation

The design models the manner in which this framework will achieve medical data interoperability and integration using XML. To conclude this work, a technology validation is offered for the purpose of evaluating the feasibility of implementing the local MIRC server functionality. Much has been written about how XML simplifies distributed computing and interoperability, and MIRC's clearly stated goal is to leverage this interoperability for achieving a unified and standardized medical CDSS framework. However, before continuing with any large-scale implementation, such as that proposed for the NICU, it is essential that the usability and ease of implementation of relevant XML technologies be evaluated and analysed.

Regardless of whether the XML processing is being performed locally or in a distributed environment, several key data manipulation objectives must be achieved that will be reusable throughout the implementation, specifically:

1. Creating Standardized XML Schemas based on medical ontologies.
2. Automatically creating XML documents representing patient information based on the transformations from a relational patient database model.
3. Automatically storing XML data in a relational model based on its schema.

4. Parsing the XML-based patient data to extract specific elements and attributes.
5. Validating the XML data based on its corresponding XML schema.
6. Processing the data based on specific XML transforms (XSLTs) to achieve a CDSS-ready data format.

The sequence diagram shown in figure 5-N depicting the class interactions required for CDSS processing makes use of steps 3-6; stages 1 and 2 must be completed before CDSS processing can be offered by MIRG (stage 1), or invoked by the client (stage 2). Figure 5-Q illustrates how the above manipulations will be employed in a distributed environment handling requests from MIRG clients and CHEO's NICU; the numbering scheme used in figure 5-Q corresponds to the numbered list above. The following sections will describe the context in which this functionality will be invoked and give general implementation details. The relevant code, XML documents (including schemas and DTDs) and patient medical database standardization work is included in a separate technical report submitted to MIRG.

Figure 5-Q Implementation Overview

5.4.1 Standardizing Medical Databases Based on the UMLS

The Unified Medical Language System's (UMLS) metathesaurus provides a user interface for standardizing medical terminology. To test this standardization, MIRG's DECH database, from the Doctor's Everett Chalmer's Hospital, containing 1492 unique patients, was used. Of the 26 concepts in the model, only three could not be mapped to any UMLS metathesaurus concept, all other terms were mapped to a UMLS-determined Concept Unique Identifier (CUI) value. CUIs make medical terminology internationally significant since the UMLS provides mappings to many different languages. The majority of the attributes are boolean values, indicated by 1 or -1, others represent discrete or string values. The concept mappings and complex data types, including acceptable data ranges, are shown in table 5-C.

It is important to realise that while a CDSS such as the ANN will only process normalized and categorical values ranging between -1 and 1, tools such as the CBR can manipulate textual data. Therefore, while the traditional means of pre-processing for the ANN is to employ a relevant categorisation scheme for non-boolean data to normalise all data input, this could result in the loss of relevant information for CBR processing. The solution is to represent all medical data in standardized format and to employ different pre-processing transforms (XSLT stylesheets) depending on which CDSS is requested.

Table 5-C UMLS-Standardized Medical Database and Schema Information

Original	New(UMLS)	CUI	Schema range	Complex data type
AD_FROM	Location from admission	-----	Non-specified	string
AD_DX1	diagnosis	C0011900	Non-specified	string
AD_DX2	diagnosis	C0011900	Non-specified	string
AD_DX3	diagnosis	C0011900	Non-specified	string
AGE	Age,NOS (yrs)	C0001779	0-200	nonNegativeInteger
CHRONIC	Chronic	C0205191	0=no 1=yes	string
EMER_SRG	Emergency surgery prior to admission?	-----	0=no 1=yes	boolean
POST_OP	Surgery prior to admission?	-----	0=no 1=yes	boolean
SEX	Gender	C0079399	-1=female 1=male	integer
TEMP	Rectal temperature	C0489749	0.00- 100.00	decimal
MAP	Mean arterial pressure	C0520852	1-200	positiveInteger
HR	Heart rate	C0018810	1-200	positiveInteger
RESP	Respiration Rate (RR)	C0518832	1-100	positiveInteger
FIO2	Inspired fraction of oxygen	C0428167	0.00-10.00	decimal
PO2	Partial pressure of oxygen in blood	C0428168	Non-specified	positiveInteger
PH	Arterial PH	C0003840	1-10	decimal
NA	Serum sodium level (mMOL/L)	C0587356	100.00-150.00	decimal
K	Serum Potassion result (mMOL/L)	C0543465	1.00-10.00	decimal
CREAT	Serum creatinine level (micromol/L)	C0600061	0.00-1000.00	decimal
HEMA	Hematocrit level	C0518014	0.00-5.00	decimal
WBC	White blood cell count laboratory result Total/mm3 in 1000's	C0427512	0.00-100.00	decimal
GCS	Glasgow coma score	C0424544	1-30	positiveInteger
DEATH	Cessation of life	C0011065	0=no 1=yes	boolean
DAYS_CNC	Duration of concentrated nursing care stay (duration)	C0449238	Non-specified	unsignedShort
DAYS_ICU	Duration of intensive care stay (ICU)	C0021708	Non-specified	unsignedShort
VENTHRS	Ventilation hours	C0035203	Non-specified	unsignedShort

5.4.2 XML Enabling the Data: XML Documents, Schemas and DTDs

The standardized UMLS terms shown in table 5-C form the basis for creating an XML document, based on XML 1.0 [W3C XML 2000], and its XML schema definition, based on the XML Schema Language Specification 1.0 [W3C Schema 2000]. Two schemas were created to validate this document (identified by an .xsd extension); one for human display purposes, which describes a conceptual structure where nodes are grouped based on UMLS semantic types, such as: admission, careGiven, etc. The other schema contains only one level of element nesting and is intended for use in CDSS processing where hierarchical data is not required.

XML Schema 1.0 allows the specification of complex data types such as string, numeric types, time-related types and XML 1.0 tokenized types (ENTITY, ID, etc..) [Chase 2001], as well as allowing the user to specify the acceptable range of attribute values. For example the standardized variable Gender (CUI C0079399) can be constrained to -1 or 1 via the statement:

```
<xs:element name="GENDER" ID="C0079399" type="integer"
minOccurs="-1" maxOccurs="1" nullable="1"/>
```

XML Schema 1.1 is intended to fix problems in version 1.0 and is still under development. Unfortunately some XML processing is not fully supported by XML Schema 1.0. In particular certain Java tools for XSLT processing were not compatible with version 1.0. For this reason, the above schemas were also represented by Document Type Definitions, which lack the explicit types and value constraints of the XML Schema but were more easily implemented. Based on a web-based review it appears that many users encountered problems validating schemas using the Xerces parser. Microsoft's XML 4.0 XML Parser was most often recommended as an alternative, but this requires a

C++ or Visual Basic coding environment, which is not desirable. This problem is expected to be resolved shortly, and more powerful XML schemas would better serve a clinical implementation.

5.4.2.1 Parsing and Validating XML Data

XML documents received from both internal and external sources must be validated prior to medical processing to ensure they match the CDSS' data constraints. To validate an XML document, based on its schema, the XML document's header must specify the schema used for validation. In MIRG where the schemas are locally stored this is a simple matter of appending:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema>
<xs:schema xmlns:xs="DECH.xsd">
```

to the start of each document, thus DECH.xsd is used to validate DECH.xml.

Initially the MIRG server's application logic must parse the received XML document to extract all information required for CDSS processing. For example, the following code segment is used to determine the requested CDSS identifier:

```
// Parsing to extract user parameters:
Document doc =parser.getDocument();
NodeList cdss = doc.getElementsByTagName("CDSS-type");
int tag1 = cdss.getLength();
String CDSS_ID;
    for (int i=0; i< num1; i++)
    {
        CDSS_ID =
        cdss.item(i).getChildNodes().item(0).getNodeValue();
    }
```

After retrieving all user CDSS processing parameters, validation is performed based on either the DTD or schema dependent on the document's specification. In both

instances this involved the creation of a new DOM parser, as shown in the following code excerpt:

```
// Validating DOM constructor:
public validatingDOM(String patientData) {
    // Create a new Xerces DOM Parser:
    DOMParser parser = new DOMParser();
    // Turn Validation on
    try {
        parser.setFeature("http://apache.org/xml/features/validatio
                           n/schema", true);
    } catch (SAXNotRecognizedException e) {
        System.err.println (e);
    } catch (SAXNotSupportedException e) {
        System.err.println (e);}
    // Register Error Handler
    parser.setErrorHandler (this);
    // Parse the Document and traverse the DOM
    try {
        parser.parse(patientData);
        Document document = parser.getDocument();
        traverse (document);
        System.out.println("Document validated.");
    } catch (SAXException e) {
        System.err.println (e);
    } catch (IOException e) {
        System.err.println (e);}
```

All code in this section is based on the Java 2 SDK development environment and the Java APIs for XML Processing (JAXP 1.1).

5.4.2.2 Schema Repositories

At this point it would be possible to publish the schemas defined in section 5.4.2 in public repositories, such as XML.org, BizTalk.org, or any more relevant distribution points within the medical community. Additionally, it would be advantageous to create a repository structure accessible by MIRG members and runtime applications for storing

schemas required for CDSS processing. Currently these schemas (and XML transforms) are simply being stored locally on the server; creating the secure repository structure is left as future work.

Such a repository structure will allow MIRG researchers to access and modify schemas to reflect the most recent database view. Schemas can be very useful in sharing results from CDSS training. For example, based on the most significant variables obtained following ANN weight elimination, new schemas can be created and added to the repository structure or, if desired, re-processed. e.g. weight elimination leads to new XML schemas.

The repository presents a means for all MIRG research to be shareable, portable and current. Additionally, MIRG's repository should be designed to support runtime processing, as described by Gregory: *"Repositories are evolving from distribution points to supplying schemas to run-time processes and are becoming critical aspects of portal-based systems"* [Gregory 2001].

5.4.3 Storing Data Received from XML-based Applications

To realise this system, XML documents must be stored at the server side for auditing and processing purposes. Basically, there are three different methods of storing XML documents:

- 1) Convert data to another model, such as a relational or object-oriented model.
- 2) Dual storage - where user stores data that has been converted to another model as well as storing the XML documents.
- 3) Store the XML documents in their native format using an XML enabled DBMS that provides document indexing.

To decide which method is most applicable one needs to consider the context in which the data is being used. Bourret suggests data that has a highly regular structure,

such as scientific and medical data, and is used by non-XML applications, will most likely be stored in a relational database and software will be required to transfer the data between XML documents and the database [Bourret 2001]. Bourret refers to this as data-centric data, where XML is mainly used for transport and processing rather than for display purposes.

It follows that the decision was made to transform data received at the MIRC server into a relational model to be stored for future auditing purposes. Kim explains that using a relational model allows the XML document to be mapped directly into the relational database server. *“The use of a relational database that maps all the tags and fields of the XML document into relational tables and columns allows for all the information in the XML document to be stored and retrieved in full context”* [Kim et al. 2002]. Additionally, *“this XML mapped database allows for complex and precise database searches using SQL and stores information in a very efficient way for machine interpretation”* [Bourret 2001].

The relational database system employed was the open source MySQL DBMS and the connection to this database was achieved via Java Database Connectivity (JDBC), a vendor independent method for accessing databases using Java. In this instance the choice of SQL DBMS is irrelevant as long as the appropriate database driver is referenced. The XML to relational conversion was more complex, time consuming and code-intensive than previous XML-based parsing, processing and validating functionality. The sole purpose of including information regarding the XML to relational transformation is to provide a point of reference for the feasibility of the implementation to aid future MIRC students involved in this work- the level of detail of the technology

validation is not deep enough to offer new insights into the storage and transformations of XML data, which is suggested as possible future work. It is hoped that employing standards such as those currently being developed in the ISO 9075, part 14 SQL/XML specifications will simplify this conversion.

At the NICU client-side, repository data received from medical devices must be converted to XML prior to transmission. This is based on transformation code that allows database information to be automatically converted to XML, based on referenced XML schemas. However, the NICU (or any other generic CDSS-client) does not just use XML as a transport mechanism, it is also used to display CDSS results in the browser. For this reason, implementing a DBMS that stores XML data in native format at the client would be advantageous in terms of reducing the number of data interfaces, and should be considered as future work.

5.4.4 Integration with MIRG's Clinical Decision Support Systems

As discussed in previous sections, the W3C's 1999 Recommendation for XSLT [W3C XSLT 1999] will be used to create an interoperable interface between MIRG's CDSSs. Such interfaces enable 'transparent' data-sharing of standardized XML data for processing by multiple CDSS applications. XSLT transforms were created for converting DECH.xml into a format that can be processed by either the ANN or the CBR.

Additionally, the results emitted from both the ANN and the CBR must be transformed into XML format and displayed on the client's browser. This requires another set of stored transforms, one for each individual CDSS and one for each logical combination of outputs achieved via hybrid CDSS processing. The correct transformations are determined based on the application logic of MIRG's servers, as shown in the sequence diagram in figure 5-N. In order to complete the integration with

MIRG's CDSSs, code must be written to implement the 'CDSS Interface' class proposed in the design section. This would allow the MIRG Application Server subsystem to automatically invoke clinical decision support; currently, CDSS invocation is a manual process.

It is important to note that each stage of the transformations described in this section must be rigorously validated to ensure that errors are not introduced as the data undergoes one or several transformations. The system will be tested on known outcomes to validate the accuracy of the CDSS results.

5.4.5 Conclusions Based on Technological Validation

Based on this small-scale implementation, utilized to validate the framework's underlying technologies, it can be concluded that XML is an effective mechanism for achieving interoperability. This conclusion is obviously subject to review based on future work in XML messaging, which will be required when the MIRG web server is created.

However, due to XML's status as a relatively new technology, there were several problems that deserve mention. The most noticeable of these is a direct result of XML's open-source policy: a proliferation of XML tools and libraries. Because many of these tools are still being developed, the XML programmer will often be faced with a situation where code that works with one parser or library does not work with another. Additionally, Internet Explorer was the only browser that supported the current XSLT recommendation. Fortunately, these problems are well documented and discussed in sites such as that offered by XML.org.

Another obstacle to quick development is the relatively time consuming problem of creating code for converting XML between various formats. Fortunately, this is only time-consuming for the original developer and can be automated for future users.

Moreover, conversion tools are being improved and developed and many of these are open source and publicly available.

As future work it would be interesting to investigate the possibility of employing the Unified Modeling Language to integrate different object models with XML. Tools are currently emerging for converting between XML and Java objects as well as other formats. The problem is that each format contains some information that the others do not. Java and other programming languages offer logical, data and behavioural control; XML provides data description; and Interface Definition Languages can represent data and behaviour. Advocates of UML are proposing that it could potentially supply the missing information by representing a given model and providing the information, to map this model into different programming languages and into XML, based on the model's XML Schema. UML currently supports representations and conversions for XML DTDs, and the Object Management Group provides a means of exchanging UML models using the XML Metadata Interchange Format (XMI). XMI provides capabilities for sharing objects using XML, via the transfer of the application's UML model to XML documents and DTDs [Anagnostaki *et al.* 2001].

Chapter 6 Conclusion

6.1 Concluding Remarks

This thesis developed a flexible medical framework for integrating clinical decision support systems and achieving medical data interoperability, employing the eXtended Markup Language for standardized information interchange. Throughout this thesis it has been emphasised that MIRG's goal is not to work in isolation but to develop a sustainable framework that can be integrated into more global contexts and handle sophisticated future clinical decision support initiatives, as well as supporting other medical applications. Future integration into an interoperable hospital information system was a key architectural driver.

It is MIRG's objective to achieve corroboration within the healthcare and medical informatics community on the long-term objectives and visions developed in this thesis. To meet this objective, this work focused on defining the framework's conceptual and contextual scope based on the higher rows of the Zachman Framework; this was predicated by the belief that organizations cannot achieve consensus at the technology and code levels (the lower rows) until a unified model has been defined and agreed upon at the 'upper levels'.

The thesis' goal of achieving medical data interoperability and application integration will not be realised without future standardization and integration work with associated costs. This is evidenced in a highly relevant quote from Geissbuhler, referring to the measurement of success of interoperable open-source systems:

“The success of these choices can only be measured when members start working not only with each other, but also with and for others, when the overhead due to harmonization and integration is effectively outweighed by the benefits of integration, scalability and coherence” [Geissbuhler et al. 2001].

6.2 Thesis Contributions

The thesis contributions have been categorised as (1) contributions to knowledge and (2) practical contributions. Within each category the contributions are ordered from most to least significant.

6.2.1 Contributions to Knowledge

- 9) Designed a medical informatics framework for integrating diverse Clinical Decision Support Systems (CDSSs) and for interoperable and unified medical information exchange in a web-based environment, based on standardized open access to medical metadata.
- 10) Developed the framework’s unifying vision, based on physician input and hospital information system trends, of an open source medical architecture that integrates generic medical applications. Additionally, the framework’s architecture is designed to support future information interchange and remote processing of real-time patient data received from the Children’s Hospital of Eastern Ontario’s Neonatal Intensive Care Unit (CHEO’s NICU) online data repository and from other generic Hospital Information Systems (HISs).
- 11) Created a medical business model identifying generic and reusable medical informatics requirements (represented by UML use cases) in the health care sector by combining medical architectural goals presented in the literature with those of this thesis’ framework. The purpose of this model is to place MIRG’s current work in a

- global context within the medical open source community and to clearly illustrate future research directions.
- 12) Suggested and analyzed choice of XML as an enabling technology to achieve CDSS semantic and logical application integration. Demonstrated that the means in which XML has been added to the framework's fundamental interchange components is unique in the field of medical informatics, where XML is primarily used as a method for representing and exchanging the Electronic Patient Record.
 - 13) Formally presented the architectural model for the above framework based on the Rational Unified Process and the Unified Modeling Language.
 - 14) Identified components of the framework's long-term architecture to be implemented as future work within the Medical IDEAS Research Group. These 'work packets' will be assigned as future student projects, the completion of which will result in the realisation of MIRG's vision of interoperable and integrated hybrid clinical decision support systems.
 - 15) Thesis work resulted in paper entitled 'Design of a Health Care Architecture for Medical Data Interoperability and Application Integration' by Catley and Frize. Accepted for publication in the EMBS-BMES 2002 (IEEE Engineering in Medicine and Biology Society) annual conference.
 - 16) Thesis work resulted in paper entitled 'Overcoming Medical Standardization Barriers with XML' by Catley and Frize. Accepted for publication in the 12th International Conference for Women Engineers and Scientists 2002.

6.2.2 Practical Contributions

- 1) Assessed choice of XML technologies for representing, processing and transforming medical data in a small-scale prototype implementation.

- 2) Standardized MIRG medical data using the Unified Medical Language System's Knowledge Source Server.
- 3) Created XML schemas for the standardized data using the XML Schema Language and XML Document Type Definitions.
- 4) Provided transforms for converting XML into relational model and vice versa. The complexity of the work generated the suggestion that future implementation take advantage of emerging XML-based repository structures for storing XML documents in their native format. These technologies could be leveraged at both the server and HIS client side.
- 5) Developed eXtended Stylesheet Language Transforms to transform patient data, represented by XML documents, into a format that can be processed by specific clinical decision support systems. This enabled common processing of standardized data by all MIRG's CDSSs.
- 6) Proposed providing physicians with an integrated XML-based result, compiling output from diverse semantically integrated CDSSs.

6.3 Future Work

During the course of this work new ideas were generated and system implementation objectives were identified that could not be pursued during the time frame of this thesis. Future work can be categorised as further XML-based research, clinical NICU implementation, clinical decision support developments or further open source and medical standardization efforts.

Further XML-based Research

- Implementing and evaluating database management techniques for storing XML documents in their native format, as compared to

employing the XML to relational transformations offered by the ISO 9075, part 14 SQL/XML specifications.

- Investigating benefits of employing the Unified Modeling Language (UML) for modeling XML schemas and DTDs and utilising XMI as a standard means of transmitting these documents. Suggested benefits would be to facilitate the translation of XML documents into other formats.
- Studying inter-relationships between UML and XML, specifically, generating XML Schemas from UML class diagrams.
- Automatically mapping Java objects to XML schemas.
- Evaluating different XML messaging standards to determine if they meet the framework's messaging requirements.
- Evaluating the XML Security Suite to determine if it can provide adequate security for information exchange of confidential patient data.
- Investigating the concept of minimal data sets and their relationship with XML metadata, e.g. employing XML to realise medical datasets where redundancy is controlled.
- Building automated user interfaces based on XML Schemas.
- Becoming involved with the push towards the 'semantic web' and its applicability in the field of clinical decision support for enabling software agents to carry out sophisticated CDSS tasks based on semantics encoded into medical web pages. XML is one of the semantic web's enabling technologies; the Resource Description Language (RDF) is the other. Developing and exploring RDFs for the CDSS domain is another element of interest.

Clinical NICU Implementation

- Integrating this framework with CHEO's NICU real-time data repository, this involves receiving and processing data in real-time (meeting hard real-time requirements) via MIRG's application server.
- XML-enabling the NICU's data repository such that conversion to XML is automatically performed based on MIRG's standardized schemas. A native XML-based repository at both the server and client side is an option for consideration.
- Actually deploying neural networks and case-based reasoning tools in the NICU and evaluating the long-term impacts on physician decision-making.
- Integrating results from MIRG's CDSSs into the patient's Electronic Patient Record stored within the Hospital Information System.

- Creating specific threshold levels for each specific CDSS outcome based on physician's inputs and integrating these thresholds into the CDSSs to generate real-time alerts when needed.
- Incorporating wireless technologies for the transmission of XML-based patient alert data to physicians' wireless devices.
- Implementing automatic CDSS invocation by the client based on data in patient repository. This would also require the implementation of acceptable thresholds for the repository data (based on medical device data).
- Implementing medical data security (based on concerns identified in this thesis).

Clinical Decision Support System Developments

- Migrating the ANN from a research to clinical setting where it would be used to process individual patient data.
- Incorporating the Replace Missing Values use case, defined in section 5.1, into the CDSS framework once the effectiveness of the proposed techniques for replacing missing values have been validated.
- Collecting input from physicians regarding acceptable system latency and modifying the framework and CDSS tools (as required) to meet these demands, e.g. utilising sufficient system resources to facilitate the required degree of concurrent processing.
- Incorporating this framework into CDSS pre-processing. This will involve achieving consensus and support from other MIRG researchers.
- Conducting a complete study of the strengths, limitations, and possibilities of hybrid (logically integrated) Clinical Decision Support Systems.
- Employing this framework as a basis for implementing a hybrid CDSS and subsequently evaluating whether the framework adequately supports hybrid CDSSs.
- Creating software agents to search other medical data repositories to obtain data for processing purposes.
- Improving CDSS tools based on benchmarking input received from external users, such as affiliated research organizations.

Open Source and Medical Standardization Efforts

- Sharing this thesis' proposed medical framework at the Open Source Health Care Alliance's annual forum, and improving the framework based on peer reviews.

- Publishing MIRG's medical schemas in XML-based medical standardization portals for discovery purposes within the medical community.
- Following the Unified Medical Language System's progress in terms of XML enabling their metathesaurus. Once this has been accomplished, MIRG's database standardization process based on accepted medical ontologies could possibly be automated (to a certain extent - it is unlikely that any database standardization process could be completely automated).

Appendix A Unified Modeling Language Overview

Figure A-A Use Case Relationships

Note: In the <<include>> relationship between Use Case 1 and Use Case 3, Use Case 3 is an abstract use case (shown in *italics*) as it is never directly invoked by an actor, but only via another use case.

Figure A-B Activity Diagram

Note: In an activity diagram the path taken after a decision point depends on the guard condition on the different transitions. In this case Activity 3 is executed if guard 2 evaluates to true and Activity 2 is executed if guard 1 is true.

The fork shows the beginning of two parallel execution paths that join together to execute Activity 6.

Figure A-C Class Diagram

Note: The fact that ClassE inherits from ClassB means that ClassE possesses all of ClassB's attributes and attributes and can also specialize Class B's functionality by creating new attributes and operations of its own.

Figure A-D Class Stereotypes

Figure A-E Sequence Diagram

Figure A-F Package Diagram

Note: Both Package 2 and Package 3 depend on Package 4.

	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	Why
SCOPE (CONTEXTUAL)	List of Things Important to the Business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations Important to the Business	List of Events Significant to the Business	List of Business Goals/Strat
Planner	ENTITY = Class of Business Thing e.g. Semantic Model	Function = Class of Business Process e.g. Business Process Model	Node = Major Business Location e.g. Logistics Network	People = Major Organizations e.g. Work Flow Model	Time = Major Business Event e.g. Master Schedule	Ends/Mean=Major Bus. Goal/ Critical Success Factor e.g. Business Plan
ENTERPRISE MODEL (CONCEPTUAL)	Ent = Business Entity Reln = Business Relationship e.g. Logical Data Model	Proc. = Business Process I/O = Business Resources e.g. "Application Architecture"	Node = Business Location Link = Business Linkage e.g. "Distributed System Architecture"	People = Organization Unit Work = Work Product e.g. Human Interface Architecture	Time = Business Event Cycle = Business Cycle e.g. Processing Structure	End = Business Objective Means = Business Strategy e.g., Business Rule Model
SYSTEM MODEL (LOGICAL)	Ent = Data Entity Reln = Data Relationship e.g. Physical Data Model	Proc. = Application Function I/O = User Views e.g. "System Design"	Node = I/S Function (Processor, Storage, etc.) Link = Line Characteristics e.g. "System Architecture"	People = Role Work = Deliverable e.g. Presentation Architecture	Time = System Event Cycle = Processing Cycle e.g. Control Structure	End = Structural Assertion Means = Action Assertion e.g. Rule Design
TECHNOLOGY MODEL (PHYSICAL)	Ent = Segment/Table/etc. Reln = Pointer/Key/etc. e.g. Data Definition	Proc. = Computer Function I/O = Screen/Device Formats e.g. "Program"	Node = Hardware/System Software Link = Line Specifications e.g. "Network Architecture"	People = User Work = Screen Format e.g. Security Architecture	Time = Execute Cycle = Component Cycle e.g. Timing Definition	End = Condition Means = Action e.g. Rule Specification
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	Ent = Field Reln = Address	Proc. = Language Stmt I/O = Control Block	Node = Addresses Link = Protocols	People = Identity Work = Job	Time = Interrupt Cycle = Watchline Cycle	End = Sub-condition Means = Step
Sub- Contractor							

Figure B-A Zachman Framework Enterprise Architecture™ [Zachman]

References

- [Ahn *et al.* 2001]
Ahn C, Yunmook N, Seunghun P, Jungsun K. An Integrated Medical Information System Using XML. *Lecture Notes in Computer Science* 2001; 2105:307-322.
- [Alaoui *et al.* 2000]
Alaoui A, Levine B, Cleary K, Mun SK. Development of a Secure Medical Research Environment. *Proc. of the IEEE EMBS International Conference on Information Technology Applications in Biomedicine* 2000;44-49.
- [Anagnostaki *et al.* 2001]
Anagnostaki A, Pavlopoulos S, Koutsouris D. XML and the VITAL Standard: The Document-Oriented Approach for Open Telemedicine Applications. *Proc. of Medinfo* 2001;77-81.
- [Anderson *et al.* 2000]
Anderson JG, Balas E, Bates DW, Kuperman GJ, Evans R, Winkelstein P. Preventing Medical Errors: Technical, Ethical, and Social Issues. *Proc. AMIA Symposium* 2000; www.amia.org/pubs/symposia/D200086.PDF ;last access June 11, 2002.
- [Anderson *et al.* 1997]
Anderson JG, Aydin CE. Evaluating the Impact of Health Care Information Systems. *International Journal of Technol. Assess Health Care* 1997;13:380-93.
- [Araki *et al.* 2000]
Araki K, Ohashi K, Yamazaki S, Hirose Y, Yamashita Y, Yamamoto R, Minagawa K, Sakamoto N, Yoshihara H. Medical Markup Language for XML-Based Hospital Information Interchange. *Journal of Medical Systems* 2000;24(3):195-211.
- [Ball *et al.* 2001]
Ball M J, Douglas J V, Lillis J. Health Informatics: Managing Information to Deliver Value. *Proc. of Medinfo* 2001;305-308.
- [Bertino & Catania 2001]
Bertino E, Catania B. Integrating XML and Databases. *IEEE Internet Computing* Jul-Aug 2001;5(4):84-88.
- [Bonifati *et al.* 2001]
Bonifati A, Ceri S, Paraboschi S. Pushing Reactive Services to XML Repositories Using Active Rules. *Commun. of ACM* 2001;633-641.
- [Booch *et al.* 1998]
Booch G, Rumbaugh J, Jacobson I. *The Unified Modeling Language User Guide*. Reading, Mass., Addison-Wesley, 1998.
- [Bourret *et al.* 1999]
Bourret R, Bornhovd C, Buchmann A. A Generic Load/Extract Utility for Data Transfer Between XML Documents and Relational Databases. TR-DVS99-1, DVS, Dep. CS, Darmstadt University of Technology, Germany, Dec. 1999.

[Bourret 2001]

Bourret R. XML and Databases, XMLDBMS;
<http://www.bourret.com/xml>, last access June 2, 2002.

[Brandt 2000]

Brandt M. Health Informatics Standards: A User's Guide. Journal of AHIMA
April 2000;71(4):39-43.

[Canfield & Sorace 2000]

Canfield K, Sorace J. Mapping XML Documents into Databases: A Data-Driven
Framework for Bioinformatic Data Interchange. Proc. of AMIA Symposium
2000;121-125.

[Chase 2001]

Chase, N. XML and Java from Scratch. Indianapolis, Indiana, QUE Publishers,
2001.

[Chester 2001]

Chester T. Cross Platform Integration with XML and SOAP. IT Proc. Sept-Oct
2001;26-34.

[CHIME 1999]

Centre for Health Informatics & Multiprofessional Education (CHIME): The
SynEx Project. Royal Free & University College Medical School, University
College London, 1999; <http://www.chime.ucl.ac.uk/HealthI/SynEx/>, last access
May 30, 2002.

[Cover 2001]

Cover R. The XML Cover Pages: Bioinformatic Sequence Markup Language
(BSML) 2001. <http://xml.coverpages.org/bsml.html>, last access May 30, 2002.

[Cover 2002]

Cover R. The XML Cover Pages: Chemical Markup Language (CML). 2002;
<http://xml.coverpages.org/cml.html>, last access May 30, 2002.

[Crichton *et al.* 2001]

Crichton D, Kincaid H, Downing GJ, Srivastava S, Hughes JS. An Interoperable
Data Architecture for Data Exchange in a Biomedical Research Network. Proc.
14th IEEE Symposium on Computer-Based Medical Systems 2001;65 –72.

[Dolin *et al.* 2001]

Dolin RH, Boles M, Dolin R, Green S, Hanifin S, Hochhalter B, Inglesis R, Ivory
M, Levy D, Nadspal K, Rae M-A, Rucks C, Snyder A, Stibolt T, Stiefel M, Travis
V. Kaiser Permanente's "Metadata-Driven" National Clinical Intranet. Proc. of
Medinfo 2001;319-323.

[Dugas *et al.* 2001]

Dugas M, Hoffmann E, Janko S, Hahnewald S, Matis T, Uberla K. XML-Based
Visual Data Mining in Medicine. Proc. of Medinfo 2001;1324-1328.

[Ennett *et al.* 2001]

Ennett CM, Frize M, Walker CR. Influence of Missing Values on Artificial
Neural Network Performance. Proc. of Medinfo 2001;449-53.

[Ennett 1999]

Ennett CM. Coronary Surgery Mortality Prediction Using Artificial Neural Networks. Master's Thesis, Dept. of Electrical Engineering, University of Ottawa, Ottawa, Ontario, 1999.

[Flesca *et al.* 2000]

Flesca S, Greco S, Zumpano E. Modeling and Querying XML Data. IEEE Database Engineering and Applications Symposium 2000;275-283.

[Frize *et al.* 2000]

Frize M, Ennett CM, Charette E. Automated Optimization of the Performance of Artificial Neural Networks to Estimate Medical Outcomes. Proc. 3rd ITAB Conference (Information Technology Applications in Biomedicine) and ITIS (International Telemedical Information Society), 2000;168-73.

[Frize *et al.* 1997]

Frize M, Trigg HCE, Stevenson M, Solven FG. Decision-Support Systems for Critical Care. Proc. of the AMIA Conference, Nashville, Oct. 25-30, 1997; <http://www.sce.carleton.ca/faculty/frize/pubs/amia97.htm>.

[Frize *et al.* 1996]

Frize M, Solven FG, Stevenson MH, Nickerson BG. Computer-Assisted Decision Support Systems for Medical Applications. Proc. of the 10th International Conference of Women Engineers and Scientists, Budapest, Hungary, October 8-12, 1996;91-93.

[Frize *et al.* 1995]

Frize M, Solven FG, Stevenson MH, Nickerson BG, Buskard T, Taylor K. Computer-Assisted Decision Support Systems for Patient Management in an Intensive Care Unit. Proc. of Medinfo 1995;1009-1012.

[Gallagher & Carnahan 2000]

Gallagher L, Carnahan L. A General Purpose Registry/Repository Information Model. Information Technology Laboratory, National Institute of Standards and Technology, 2nd draft, October 23, 2000.

[Geissbuhler *et al.* 2001]

Geissbuhler A, Lovis C, Lamb A, Spahni S. Experience with an XML-HTTP-Based Federative Approach to Develop a Hospital-Wide Clinical Information System. Proc. of Medinfo 2001;735-739.

[Gomaa 2000]

Gomaa H. Designing Concurrent, Distributed and Real-Time Applications with UML. Addison Wesley Longman, 2000.

[Gregory 2001]

Gregory A. XML Schema Design for Business-to-Business e-Commerce. XML Europe 2000; <http://www.gca.org/papers/xmleurope2000/author/s21-01auth1.html>, last access May 30, 2002.

[Gritzalis *et al.* 2001]

Gritzalis S, Gritzalis D, Moulinos C, Iliadis J. An Integrated Architecture for Deploying a Virtual Private Medical Network Over the Web. Medical Informatics and the Internet in Medicine 2001;26(1):49-72.

[HL7 2002]

HL7 Standards: HL7 Version 3 (draft). Health Level Seven, Ann Arbor, MI;
<http://www.hl7.org/library/standards.cfm>, last access May 30, 2002.

[Huff *et al.* 1998]

Huff SM, Bidgood DW, Cimino JJ, Hammond WE. A Proposal for Incorporating Health Level Seven Vocabulary in the UMLS Metathesaurus. *Journal of the AMIA*, Pt Supplement 1998;800-804.

[IBM XML 2002]

IBM: XML Zone; <http://www-106.ibm.com/developerworks/xml/>, last access May 30, 2002.

[IEEE 1999]

IEEE's Recommended Practice for Architectural Description. IEEE Architecture Working Group, 1999;
<http://www.firstmark.ca/ContentDocs/architectural%20docs/IEEE1471%20Arch%20Standard/IEEEp1471recommendedpracticearchitectures1.pdf>, last access June 3, 2002.

[IEEE 2002]

Workshop Report on Integrating XML & Distributed Object Technologies;
http://jeffsutherland.org/xml/IEEE_XML_Report_Draft.htm, last access May 30, 2002.

[Ingenerf *et al.* 2001]

Ingenerf J, Reiner J, Seik B. Standardized Terminological Services Enabling Semantic Interoperability between Distributed and Heterogeneous Information Systems. *International Journal of Medical Informatics* 2001;64:223-240.

[Jacobson *et al.* 1999]

Jacobson I, Booch G, Rumbaugh J. *The Unified Software Development Process*. Addison-Wesley, Upper Saddle River, NJ, 1999.

[Kahn & de la Cruz 1998]

Kahn C, de la Cruz N. Extensible Markup Language (XML) in Health Care: Integration of Structured Reporting and Decision Support. *Proc. AMIA Symposium* 1998;725-9.

[Karadimas *et al.* 2000]

Karadimas H, Hemery F, Lepage E, Roland P. DbMap: Improving Database Interoperability Issues in Medical Software Using a Simple, Java-XML Based Solution. *Proc. AMIA Symposium* 2000;408-412.

[Kasukawa *et al.* 1999]

Kasukawa T, Matsuda H, Nakanishi M, Hashimoto A. A New Method for Maintaining Semi-Structured Data Described in XML. *1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* 1999;258-261.

[Keith & Greene 2001]

Keith D F & Greene K. The Application of Intelligent Knowledge Based Systems for Risk Management and Decision Support in Obstetrics. *Bioinformatics*

Symposium: Computer-based Medical Instrumentation & Information Technologies & Technology Fair 2001;17-19.

[Kim *et al.* 2002]

Kim J, Feng D D, Cai T W, Eberl S. Integrated Multimedia Data Agent in E-Health. Conferences in Research and Practice in Information Technology, Vol 11, <http://www.cs.usyd.edu.au/~vip2001/proceedings/jinman1.pdf>, last access June 11, 2002.

[Kohn *et al.* 2000]

Kohn LT, Corrigan J, Donaldson MS. Institute of Medicine (U.S) Committee on Quality of Health Care in America. To Err is Human: Building a Safer Health System. Washington DC, Nat Acad Press, 1999.

[Kuhn & Giuse 2000]

Kuhn KA, Giuse DA. From Hospital Information Systems to Health Information Systems: Problems Challenges, Perspectives. *Methods of Information in Medicine* 2001;40:275-287.

[Lear 1999]

Lear A. XML Seen as Integral to Application Integration. *IT Proc.* Sept/Oct 1999;12-16.

[Luttgau & Bendl 2001]

Luttgau A, Bendl R. Technical Aspects of Internet-Based Knowledge Presentation in Radiotherapy. *Medical Informatics and the Internet in Medicine* 2001;26(4):265-281.

[Manickam & Abidi 2001]

Manickam S, Abidi A. Extracting Clinical Cases from XML-Based Electronic Patient Records for Use in Web-Based Medical Case Based Reasoning Systems. *Proc. of Medinfo* 2001;643-647.

[McDonald 1997]

McDonald CJ. The Barriers to Electronic Medical Record Systems and How to Overcome Them. *Journal of AMIA* 1997;4:213-21.

[Megginson 2000]

When XML Turns Ugly. Megginson Technologies March 2000; <http://www.megginson.com/ugly/index.html>, last access June 2, 2002.

[Mehra *et al.* 2000]

Mehra V, Walker R, Brashear J, Tavorath M. Designing a Flexible Services-Based Architecture for Internet Applications. *Commun. of ACM* 2000;31-32.

[Muller *et al.* 2001]

Muller M, Ganslandt T, Eich H P, Lang K, Ohmann C, Prokosch H. Towards Integration of Clinical Decision Support in Commercial Hospital Information Systems Using Distributed Reusable Software Components. *International Journal of Medical Informatics* 2001;64:369-377.

[OASIS 2002]

OASIS; www.oasis-open.org, last access May 30, 2002.

[OASIS-2 2002]

OASIS Technical Work Overview. <http://www.oasis-open.org/committees/index.shtml>, last access May 30, 2002.

[Okstein 1999]

Okstein C, XML: A Key Technology for Sharing Clinical Information. MD Computing 1999;16(9):31-33.

[OSHCA 2001]

The OSHCA Charter. Minoru Development Corporation, 2001;
<http://www.oshca.org/charter.html>, last access May 30 2002.

[OMG 1999]

Momentum Increases Behind CORBA and XML Integration. The OMG and OASIS Exchange Memberships. Object Management Group, 1999;
http://cgi.omg.org/news/pr99/9_21.html, last access June 2, 2002.

[Pokorny 2000]

Pokorny J. XML Functionally. International Database Engineering and Applications Symposium 2000;266-74.

[Pons *et al.* 1999]

Pons A, Millet J, Gijarro E, Mainteiga M. Medical Database Migration Using New XML Standard. Proc. of IEEE: Computers in Cardiology 1999;26:93-96

[Quatrani 1998]

Quatrani T. Visual Modeling with Rational Rose and UML. Addison Wesley Longman, 1998.

[Reese]

Reese G. Database Programming with JDBC and Java. O'Reilly, 2000.

[Roy & Ramanujan 2001]

Roy J, Ramanujan A. XML Schema Language: Taking XML to the Next Level. IT Proc. Mar-Apr 2001;37-40.

[Schmidt *et al.* 2001]

Schmidt R, Montani S, Bellazi R, Portinale L, Gierl L. Case-Based Reasoning for Medical Knowledge-Based Systems. International Journal of Medical Informatics 2001;64:355-367.

[Schulz & Hahn 2001]

Schulz S, Hahn U. Medical Knowledge Reengineering-Converting Major Portions of the UMLS into a Terminological Knowledge Base. International Journal of Medical Informatics 2001;64(2):207-221.

[Seligman & Rosenthal 2001]

Seligman L, Rosenthal A. XML's Impact on Databases and Data Sharing. Proc. IEEE Computer June 2001:59-67.

[Smith 2001]

Smith DA. Data Transmission: A World of Possibilities. American Health Information Management Association, 2001;
<http://www.ahima.org/journal/cutting.edge/0105.html>, last access May 30, 2002.

[Sokolowski 1999]

Sokolowski R. Expressing Health Care Objects in XML. Proc. IEEE 8th International Workshops on Enabling Technologies: Information for Collaborating Enterprises 1999;341-342.

[Stalidis *et al.* 2001]

Stalidis G, Prentza A, Vlachos I, Maglavera S, Koutsouris D. Medical Support System for Continuation of Care Based on Web Technology. International Journal of Medical Informatics 2001;64(2):385-400.

[Taylor *et al.* 1993]

Taylor KB, Nickerson BG, Frize M, Solven FG, Dunfield V. Use of Case-Based Reasoning to Assess Patient Management in an Intensive Care Unit. Proc. of Joint Conf. COMP and CMBES, Ottawa 1993;248-249.

[Tidwell 2000]

Tidwell, D. The XML Security Suite: Increasing the Security of e-Business. IBM developerWorks April 2000; <http://www-106.ibm.com/developerworks/webservices/library/xmlsecuritysuite/index.html>, last access May 30, 2002.

[UK 1996]

Confidential Enquiry into Stillbirths and Deaths in Infancy: 5th Report, 1 January–31 December 1996: 1–109. London, Maternal and Child Health Research Consortium, 1998.

[UMLS 2002]

Unified Medical Language System (UMLS): 2002 UMLS Documentation. U.S. National Library of Medicine, National Institutes of Health, Dept. of Health & Human Services, Bethesda, MD, 2002; <http://www.nlm.nih.gov/research/umls/UMLSDOC.HTML>, last access May 30, 2002.

[W3C Encr. 2002]

XML Encryption WG. W3C, 2002; <http://www.w3.org/Encryption/2001/>, last access May 30, 2002.

[W3C Schema 2000]

W3C Architecture Domain: XML Schema Version 1.0, 2000; <http://www.w3.org/XML/Schema>, last access May 30, 2002.

[W3C SOAP 2000]

W3C Simple Object Access Protocol 1.1, 2000; <http://www.w3.org/TR/SOAP/>, last access June 2, 2002.

[W3C XML 2000]

W3C Architecture Domain: Extensive Markup Language (XML) Version 1.0, 2000; <http://www.w3.org/XML/>, last access June 11, 2002.

[W3C XSLT 1999]

W3C XSL Transformations (XSLT) Version 1.0, 1999; <http://www.w3.org/TR/xslt>, last access May 30, 2002.

[Walker 2001]

Walker R. Personal Correspondence- MIRG Research Meeting. January 2001.

- [Wang 1999]**
Wang, C.X. Security Issues to Tele-Medicine System Design. Proc. of Southeastcon '99 (IEEE) 1999;106-9.
- [Watson 2001]**
CORBA and XML; Conflict or Co-operation? Object Management Group, 2001;
<http://www.omg.org/news/whitepapers/watsonwp.htm>, last access June 3, 2002.
- [Widom 1999]**
Widom J. Data Management for XML: Research Directions. IEEE Data Engineering, Sept. 1999;42-52.
- [Wilson *et al.* 1995]**
Wilson RMcL, Runciman WB, Gibberd RW, et al. The Quality in Australian Health Care Study. Med J Aust 1995;163:458-471.
- [Wolff *et al.* 2001]**
Wolff A, Mluddek V, van der Haak M, Bork W, Bulze-bruck H, Schmucker P, Wannenmacher M, Haux R. Using the Extensible Markup Language (XML) in a Regional Electronic Patient Record for Patients with Malignant Diseases. Proc. of Medinfo 2001;698-702.
- [XMLDB 2002]**
Initiative for XML Databases. The XML:DB Initiative, 2001;
<http://www.xmldb.org/>, last access May 30, 2002.
- [XML.org1 2002]**
The XML Industry Portal. XML.org, 2002; <http://www.xml.org>, last access May 30, 2002.
- [XML.org2 2002]**
The XML Industry Portal: About XML.org. XML.org, 2002;
<http://www.xml.org/xml/aboutxml.shtml>, last access May 30, 2002.
- [XML-RPC 2001]**
XML-RPC Home Page. XML-RPC.com. November 2001;
<http://www.xmlrpc.com/>, last access June 2, 2002.
- [Xu *et al.* 2001]**
Xu Y, D'Alessio L, Jaulent M-C, Sauquet D, Spahni S, Degoulet P. Integrating Medical Applications in an Open Architecture Through Generic and Reusable Components. Proc. of Medinfo 2001;63-67.
- [Yackel 2001]**
Yackel TR. How the Open-Source Development Model Can Improve Medical Software. Proc. of Medinfo 2001;68-72.
- [Yale 1997]**
Yale K. Preparing the Right Data Diet for Training Neural Networks. IEEE Spectrum March 1997;64-66.
- [Zachman Int]**
Zachman JA. Enterprise Architecture: A Framework TM. Zachman Institute for Framework Advancement (810) 231-0531. Zachman International.