A Standards-based ICT Framework to Enable a Service-Oriented Approach to Clinical Decision Support

By

Salvador Rodríguez Loya

Submitted for the degree of Doctor of Philosophy
at the University of Sussex

Department of Engineering and Design,
School of Engineering & Informatics

University of Sussex
Brighton

September 2014

Summary

This research provides evidence that standards based Clinical Decision Support (CDS) at the point of care is an essential ingredient of electronic healthcare service delivery. A Service Oriented Architecture (SOA) based solution is explored, that serves as a task management system to coordinate complex distributed and disparate IT systems, processes and resources (human and computer) to provide standards based CDS.

This research offers a solution to the challenges in implementing computerised CDS such as integration with heterogeneous legacy systems. Reuse of components and services to reduce costs and save time. The benefits of a sharable CDS service that can be reused by different healthcare practitioners to provide collaborative patient care is demonstrated. This solution provides orchestration among different services by extracting data from sources like patient databases, clinical knowledge bases and evidence-based clinical guidelines (CGs) in order to facilitate multiple CDS requests coming from different healthcare settings. This architecture aims to aid users at different levels of Healthcare Delivery Organizations (HCOs) to maintain a CDS repository, along with monitoring and managing services, thus enabling transparency.

The research employs the Design Science research methodology (DSRM) combined with The Open Group Architecture Framework (TOGAF), an open source group initiative for Enterprise Architecture Framework (EAF). DSRM's iterative capability addresses the rapidly evolving nature of workflows in healthcare. This SOA based solution uses standards-based open source technologies and platforms, the latest healthcare standards by HL7 and OMG, Decision Support Service (DSS) and Retrieve, Update Locate Service (RLUS) standard. Combining business process management (BPM) technologies, business rules with SOA ensures the HCO's capability to manage its processes. This architectural solution is evaluated by successfully implementing evidence based CGs at the point of care in areas such as; a) Diagnostics (Chronic Obstructive Disease), b) Urgent Referral (Lung Cancer), c) Genome testing and integration with CDS in screening (Lynch's syndrome). In addition to medical care, the CDS solution can benefit organizational processes for collaborative care delivery by connecting patients, physicians and other associated members. This framework facilitates integration of different types of CDS ideal for the different healthcare processes, enabling sharable CDS capabilities within and across organizations.

Acknowledgements

I would like to thank my PhD supervisor, Professor Chris Chatwin, for supporting me throughout the research, for his advice and insightful discussions and suggestions. He was always willing to listen and share his knowledge. I also have to thank the members of the Informatics and Data Systems Research laboratory, Dr Rupert Young and Dr Phil Birch for their advice and suggestions.

I would like to thank my colleague Ayesha Aziz for her advice, important contributions and interesting discussions. Ayesha always was willing to stop her work in order to provide help and support in any way. I also want to thank Brandon Welch for inviting me to contribute in his fascinating research about the use of Whole Genome Data information in CDS.

I will forever be thankful to Dr Kensaku Kawamoto for letting me participate in his research group and for his tireless support and patience.

I would like to thank my parents for their unconditional support. To my father who taught me the values of honesty, integrity, responsibility, hard work and that the family always comes first. There are not enough words to thank my father for what he has done for me in my life. To my mother whose love and support made it possible to pursue my dreams for so long. She has been a source of endless inspiration. To my sister who is my best friend and confident, and who always has been my cheerleader.

I would like to thank my brother Roger for his admirable strength and generosity and for always answering my calls and been willing to talk. To my bother Ricky who has been an example of courage and perseverance. To my bother Chuy who is an example of honesty and integrity.

Table of contents

List of Figures

List of Tables

List of Publications

- S. Rodriguez-Loya, A. Aziz, and C. Chatwin, “A Service Oriented Approach for Guidelines-based Clinical Decision Support using BPMN,” *Stud. Health Technol. Inform.*, vol. 205, p. 43—47, 2014.
- S. Rodriguez, A. Aziz, and C. Chatwin, “Enabling Healthcare IT Governance: Human Task Management Service for Administering Emergency Department’s Resources for Efficient Patient Flow,” *Stud. Health Technol. Inform.*, vol. 202, p. 87—90, 2014.
- Aziz, S. Rodriguez, and C. Chatwin, “From Guidelines to Practice: Improving Clinical Care through Rule-Based Clinical Decision Support at the Point of Care,” in *Rules on the Web. From Theory to Applications SE - 13*, vol. 8620, A. Bikakis, P. Fodor, and D. Roman, Eds. Springer International Publishing, 2014, pp. 178–185.
- M. Welch, S. R. Loya, K. Eilbeck, and K. Kawamoto, “A Proposed Clinical Decision Support Architecture Capable of Supporting Whole Genome Sequence Information,” *J. Pers. Med.*, vol. 4, no. 2, pp. 176–199, 2014.
- S. Loya, K. Kawamoto, C. Chatwin, and V. Huser, “Service Oriented Architecture for Clinical Decision Support: A Systematic Review and Future Directions,” *J. Med. Syst.,* vol. 38, no. 12, pp. 1–22, 2014.
- B. M. Welch, S. Rodriguez-Loya, K. Eilbeck, and K. Kawamoto, “Clinical Decision Support for Whole Genome Sequence Information Leveraging a Service-Oriented Architecture : a Prototype,” in *AMIA 2014 Annual Symposium*, 2014.
- S. Rodriguez and Y. Rodriguez, “Arquitectura SOA Basada en Componentes de Software de Código Abierto para la Implementación de Historia Clínica Electrónica,” in *IX Congreso Internacional Informática en Salud 2013*, 2013.

Acronyms

ArchiMate – ArchiMate is an independent modelling standard for describing Enterprise Architectures. It was developed by The Open Group. ArchiMate focuses on the overview and coherence instead of the specificity and detail, as do other modelling languages such as UML (Unified Modelling Language) and BPMN.

BPM – Business Process Management is a well-established discipline that "combines knowledge from information technology and knowledge from management sciences and applies this to operational business process".

BPMN – Business Process Model and Notation is a graph-oriented language that can be used for conceptual modelling and to execute processes.

BR – It stands for Business Rule. A BR can be described as a "statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business. The business rules that concern the project is atomic – that is, they cannot be broken down further".

BRMS – Business Rules Management System is a computer systems used to manage Business Rules.

C-CDA – Consolidated Clinical Document Architecture is an XML-based specification used to transmit patient-specific medical data.

CDS – Clinical Decision Support provides clinicians, patients, or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care.

CDS Orchestrator – This is the term used for the implementation and evaluation of the various components of the architecture solution presented in this thesis. It is a name given for the collection of tools, methodologies and technologies that together enable an SOA based standards compliant CDS at the point of care.

CDSS – Clinical Decision Support System is an information system that provides CDS.

CPOE – Computerised Physician Order Entry System is a computer system that allows physicians to make prescription entries such as medication and diagnostics.

DSS – It refers to the Clinical Decision Support Service Standard developed by the Health Services Specification Program which is collaboration between HL7 and OMG.

DSS – It stands for Decision Support Service. It is a Web service specification developed by HSSP. It can receive patient data and return patient-specific assessments and recommendations. It supports payloads based on standards such as the Consolidated Clinical Document Architecture (C-CDA) or vMR.

EA – Enterprise Architecture is the organization for business process and IT capabilities reflecting the integration and standardization requirements of the firm's operation model.

EHR – Electronic Health Record is a term used to refer to the information system that manages electronic medical records.

ESB – Enterprise Service Bus is a message backbone that supports several message patterns such as Event-driven messaging, intermediate routing, asynchronous queuing, protocol bridging, data model transformation and data format transformation.

HCO – Health Care Organization is an organization that provides health care services.

HHS – It stands for the Office of the Secretary for the U.S. Department of Health and Human Services.

HIE – Health Information Exchange refers to the electronic sharing of information across health care organizations.

HIT – Health Information Technology is the application of information processing involving both computer hardware and software that deals with the storage, retrieval and use of health care information, data and knowledge for communication and decision making.

HL7 – Health Level Seven is an international authority on standards for interoperability of health information technology.

HSSP – Health Services Specification Program is a joint effort between Health Level 7 (HL7) and the Object Management Group (OMG) that focuses on service specification standards based on SOA principles.

ICT – Information and Communication Technology is a term used to refer the use of computers and other technologies for data processing.

Infobuttons – Infobuttons can be described as context-aware links to online knowledge resources that are embedded into clinical systems.

IOM – The Institute of Medicine is an independent, non-profit organization that provides advice to the government and the public on health and health care.

LIS – Laboratory Information System is an information system that manages laboratory data.

NICE – National Institute for Health and Care Excellence is an organization that provides national guidance and advice to improve health care in the United Kingdom.

OASIS – Advancing Open Standards for the Information Society is a non-profit consortium that drives the development, convergence and adoption of open standards.

OMG – Object Management Group is an international authority on computer standards for a wide range of industries.

ONC – It stands for the Office of the National Coordinator for Health Technology. ONC is part of the Office of the Secretary for the U.S. Department of Health and Human Services (HHS).

PACS – Picture Archiving and Communication System is an information system that deals with transition, storage, processing and display of medical images.

RIS – Radiology Information System manages information generated in the Radiology department.

RLUS – It stands for Retrieve, Locate, and Update Service. It is a Web services specification developed by HSSP. RLUS defines a set of interfaces that allows locating, accessing and updating health data from healthcare organizations.

SCA – Service Component Architecture is a set of OASIS standards specifically design to support the development of service oriented applications.

SDO – Standard Development Organization is term used to refer to the organizations specialised on the development of standards.

SOA – Service Oriented Architecture is a computing paradigm for organizing and consuming distributed functionalities.

SOAP – Simple Object Access Protocol is an XML-based specification for the exchange of information between applications.

TOGAF – It stands for The Open Group Architecture Framework. TOGAF is a standard developed by The Open Group. It defines a methodology for developing an Enterprise Architecture.

vMR – It stands for Virtual Medical Record. It is a specification designed for transferring data between CDS systems. It includes a set of clinical data elements required to implement CDS and defined inputs and outputs between CDS systems and healthcare information systems.

WSDL – Web Services Description Language is an XML-based interface description language used to declare Web services.

XML – eXtensible Markup Language is a standard that defines a set of rules for structuring documents.

Chapter 1

Introduction

1.1. Background

Health care relies heavily on the collection and synthesis of information. However, there is an increasing concern whether there is efficient use of information to improve quality of care [1]. The wide range of clinical knowledge, the continuous advances in Medical Science and the demanding nature of the clinical environment makes it extremely difficult for clinicians to stay up to date with the latest research and development in their respective fields. The past few years have given rise to various solutions that address the problem of managing health information. These solutions primarily involve the use of information and communication technology (ICT) and are commonly termed as Health Information Technology (HIT). An increasing number of governments around the world have adopted programs to promote the use of HIT on a national level. HIT includes technology based solutions for managing the entire healthcare enterprise starting from patient administration to complex medical imaging systems. Among these HIT innovations, a significant number of new technologies are related with Clinical Decision Support (CDS). "CDS provides clinicians, patients, or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care" [2]. Studies have shown that CDS can support clinical decision making, improve patient outcomes, reduce unnecessary mistakes and expenses, and increase efficiency [3]. Despite the demonstrated benefits that CDS can provide to Health Care Organizations (HCO), there are many barriers that prevent the adoption of CDS at scale. A major challenge is the integration of CDS interventions with the overall IT implementation for an HCO. This necessitates research into the design and architecture of the HIT implementation. There has been an increasing trend in adopting a Service Oriented Architecture (SOA) approach for implementing enterprise wide IT systems in industries like manufacturing and banking. Healthcare IT implementation requires an architecture that is able to share and exchange complex medical and non-medical information across multiple distributed environments. Hence the use of SOA for HIT is being adopted by various vendors providing technological support for healthcare service delivery. This thesis builds on the concepts of integration of healthcare enterprise IT systems. The main focus of this thesis is to present an SOA-based integration solution for the CDS implementation. The architecture is capable of supporting various features of the CDS infrastructure in a HIT and is implemented and evaluated against various case studies.

1.2. Overview of Research

This dissertation focuses on HIT, specifically the use of technology to manage clinical and non-clinical tasks for collection, extraction and presentation of patient information for Clinical Decision Support (CDS). The background study identifies various methods of defining and developing, a standard-based solution to enable a service-oriented approach to clinical decision support from the software architecture point of view. CDS systems operate in an environment that involves participation of multiple actors, like government regulations, healthcare professionals, hospitals, insurance companies, drug manufacturers etc. In order to address the complexity that arises due to the involvement of multiple actors (human and software), the research project considered the following as vital factors for the successful implementation of CDS.

1. ***Interoperability***:
   a. Support different communication protocols.
   b. Data and message transformation and enrichment.
   c. Support different Health Information Exchange (HIE) standards.
2. ***Manageability:***
   a. Allow healthcare professionals to readily design and define the orchestration logic
3. ***Scalability:***
   a. Allow addition of new functionalities with minimal effort

1.3. Research Gap

The research gaps are identified by conducting a Systematic Review of current trends in Service Oriented Architecture for Clinical Decision Support. The results of this survey is presented in Chapter 2 and Chapter 3 There is a need for an approach to orchestrate clinical tasks that result from different distributed healthcare systems and human actors in order to enable a service-oriented approach to clinical decision support. The orchestration logic should be defined using a notation readily understood by both developers and healthcare professionals. The resulting system should be system- agnostic and support healthcare standards.

"CDS Orchestrator" is the name given to a collection of tools, methodologies and technologies that together enable an SOA based standards compliant CDS at the point of care. This research focuses on the implementation and evaluation of the architecture required for a CDS Orchestrator.

1.4. The Aim of the Research

The aim of the research is to define and validate an architectural approach for a task management service to enable a service-oriented approach to clinical decision support. The approach should be able to adapt to different healthcare settings (e.g., primary care, secondary care, and tertiary care).

The following assumptions and constraints were considered during this research:

Assumptions:

1. There are several healthcare systems that do not interact with each other because of the lack of interoperability in the targeted healthcare organization.
2. Secure and reliable broadband internet service is available.

Constraints:

1. Patient information is spread among different systems within an organization or even in different organizations.
2. Physicians may work in different healthcare organizations
3. Although there are similarities between organizations, each organization is unique because of the services that it provides and the resources available.

1.5. Research objectives

The objectives of this research are listed below:

1. Identify current trends and challenges in Architecture approaches for Clinical Decision Support. This objective is achieved by conducting a Systematic review

of Service Oriented Architecture for Clinical Decision Support. (Chapter 2 and Chapter 3)

2. Enable an SOA based architecture framework that leverages WGS information to facilitate the implementation of personalized medicine and therefore improve health care. (Chapter 4)
3. Implement an SOA-based environment using CDS services built upon modelled clinical guidelines combining the business process language BPMN and business rules. (Chapter 5)
4. Develop a human task management architecture component that allows the coordination of tasks specified in workflows. (Chapter 6)

1.6. Scope of Research

Based on the identified challenges of existing approaches for integration of CDS, the scope of this research covers the HIT aspects relevant to the architecture of a task management service to enable a service-oriented approach to CDS. A systematic architecture was developed and implemented as a result of the knowledge acquired during the research. The research concerns include data integration, application integration, and technology infrastructure.

1.7. Research Questions and Strategy

Based on the problem statement described in section 1.3 the following research questions (RQ) were established:

- RQ1: How can an architecture solution that enables a service-oriented approach to clinical decision support that provides interoperability and information accessibility across healthcare systems can be developed?
- RQ2: What would be an effective architecture approach that allows the integration of architecture solution into the healthcare workflow for both clinical and non-clinical scenarios?
- RQ3: How can the architecture solution provide interoperability between different healthcare systems?

- RQ4: How can an architecture solution that enables a service-oriented approach for clinical decision support improve the quality of care?
- RQ5: How can current healthcare standards be integrated with the SOA based architecture solution?

Based on these research questions, the following proposition had been defined:

Strategy: A systematic architecture approach should be developed which enables a service-oriented approach by orchestrating clinical tasks required to collect patient information necessary to provide clinical decision support. The developed systematic architecture should guide the development of an integrated solution that addresses the issues of interoperability, manageability, and scalability.

1.8. Research Methodology

This thesis adopts a Design Science Research Methodology (DSRM) [4] and also incorporates some of the best practices identified for the information systems development industry. The main objective of the mixed research methodology is to provide answers to the research questions through the development of an architectural model and prototype implementation.

Software Configuration management

Software Configuration Management (SCM) is a well-established discipline that controls the evolution of systems and covers a wide range of aspects of the software development cycle [5]. Some of the basic aspects of SCM are software version management and build automation. In this research study, the open source tools Git (http://git-scm.com/) and Maven (http://maven.apache.org/) were used to support the development of the proposed solution. Git is a distributed version control system that focuses on speed, performance, flexibility and usability. Maven is a build automation tool that is widely used among developers and project managers.

1.9. Achievements

The research provides contributions to the issues of clinical task management for CDS in the context of service oriented architecture. The research contributions are listed below:

1. An extensive review of the theory and current implementations was performed resulting in requirements for a SOA based CDS integration solution. (Chapter 2,3).
2. A conceptual solution was established following a systematic architectural approach. The conceptual solution allows the orchestration of clinical tasks required to provide CDS. (Chapter 4)
3. A functional prototype was implemented and tested against different healthcare scenarios. The prototype supports different healthcare standards such as HL7 DSS and HL7 RLUS. (Chapter 4, 5, 6)
4. Development of a plugin for the Open Source Electronic Health Record (EHR) system Tolven, that allows the integration of the clinical decision support service OpenCDS. (Chapter 4, 5, 6)
5. Leveraging the best practices of SOA, an architecture solution for integration of various components facilitating point of care decision support was developed. (Chapter 4, 5, 6)
6. The architectural approach and prototype addresses interoperability issues by supporting several communication protocols and healthcare standards. The logic of the management of clinical tasks is exposed in a notation that is readily understood by healthcare professionals. (Chapter 4, 5, 6)

1.10. Thesis structure

The dissertation is composed of the following chapters. The chapters correspond to the DSRM steps as shown in figure. 1

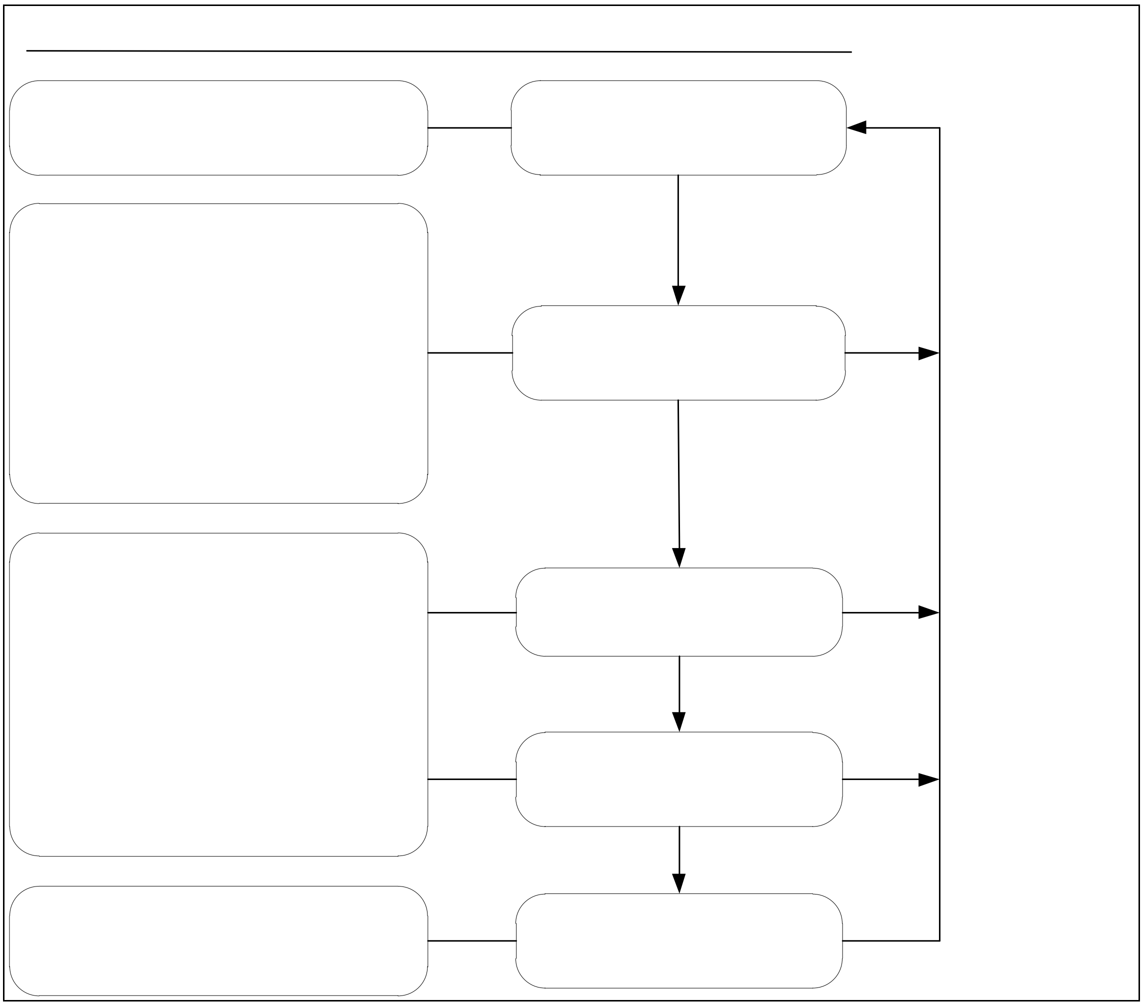


Figure 1: Overview of the research process based on the design science methodology

Chapter 2 Service Oriented Architecture for Clinical Decision Support: A Systematic Review

This chapter presents a Systematic Review that focus on identifying service oriented approaches adopted by researchers and developers during the implementation of Clinical Decision Support Systems. This Systematic review was result of collaboration between the University of Utah, USA and the University of Sussex.

Chapter 3 Evolution of Clinical Decision Support Systems: Health Information Technology, Standards and Implementation Approaches.

This chapter briefly describes the benefits of adopting HIT. It summaries the challenges and successful implementation strategies for HIT reported by the literature. It describes the concept of CDS and the benefits that it brings to Health Care Organizations (HCOs). Different implementation approaches of CDS are presented. The barriers that prevent the implementation of CDS at scale are described. Some of the most important Healthcare standards related to CDS are also presented.

The notion of Enterprise Architecture (EA) is introduced and how it is used to develop a complex systems solution is also discussed. The architectural style, Service Oriented Architecture (SOA) is defined and a review of technologies around SOA is also presented. These technologies include: Service Component Architecture (SCA), Business Process Management (BPM), Business Rules (BR) and Human Interaction Workflows. Finally the gaps identified in the literature concerned with the implementation of CDS are presented in the final section of this chapter.

Chapter 4 Whole Genome Sequence Guided Clinical Decision Support

This chapter discusses the use of Whole Genome Sequence (WGS) information to provide CDS at the point of care. Because of the rapid advances in genetics, there is a growing interest in using genetic information in clinical settings. However, due to the complexity of genomic information and the continuous changes in the interpretation of such information, there are several aspects (e.g. interpretation of Genetics database, integration with EHRs etc.), that must be considered when developing CDS based on WGS information. This chapter describes the design, implementation and evaluation of a WGS-guided CDS, which is integrated into the clinical workflow and realized by displaying alerts at the user interface of an Electronic Health Record (EHR) system. The prototype implementation is described through a clinical scenario of a hypothetical patient with high risk for Lynch Syndrome based on genetic information.

Chapter 5 A Service Oriented Approach for Guidelines-based Clinical Decision Support using BPMN and Business Rules

This chapter presents a SOA approach for developing CDS services. It describes the development and implementation of two CDS Web services based on models of clinical guidelines created combining the business process language BPMN and Business Rules. Each Web service is deployed as a Service Component Architecture (SCA) composite, which can interact with an EHR. The models are based on the clinical guidelines for Chronic Obstructive Pulmonary Disease (COPD) and preliminary diagnosis for Lung Cancer respectively.

This chapter presents the results of implementing the SOA based architecture solution in which clinical guidelines are represented as business processes. Since guidelines describe a set of activities in a specific order, business processes are used to implement them. The general purpose business process modelling language BPMN is widely used in other industries but it is relatively new in the field of clinical guideline modelling.

Chapter 6 Human Task Management

This chapter describes how business process technology can be used to coordinate interactions between human actors and information systems in healthcare organizations. It describes the implementation of a human task management module, which can be integrated to the architecture described during chapters three and four. The functionalities of this module are demonstrated through the partial implementation of the clinical guideline for managing an Acutely iII Patient in the Emergency Department developed by the National Institute for Health and Care Excellence (NICE).

Chapter 7 Conclusions and Future Directions

This chapter reviews and validates the research proposition and research questions. The prototype of the architecture proposed is analysed in terms of the challenges of CDS implementation. It discusses the main contributions and limitations of the present research. The final section presents suggestions for future research.

Chapter 2

Service Oriented Architecture for Clinical Decision Support: A Systematic Review

2.1. Introduction

This chapter presents the results of a systematic review conducted to identify Service Oriented Architecture (SOA) approaches adopted by developers and researchers during the development of Clinical Decision Support (CDS) systems. This systematic review was conducted in collaboration with Dr. Kensaku Kawamoto (Department of Biomedical Informatics, University of Utah) and Dr. Vojtech Huser,(Research Program of the National Institutes of Health Clinical Center and the National Library of Medicine). The Appendix A shows the details of methodology adopted and selected studies for this systematic review. The data extraction and quality assessment was performed by the author. Data Synthesis and Analysis were performed by the authors and included reviews from medical and health informatics professionals.

2.2. Results

The study selection process is presented in figure 2. The electronic search returned 135 unique studies. After reading the title and abstract, 42 articles were selected and read in full. Subsequently, 32 articles were included in the final review. Appendix A shows the complete list of articles included in the systematic review.

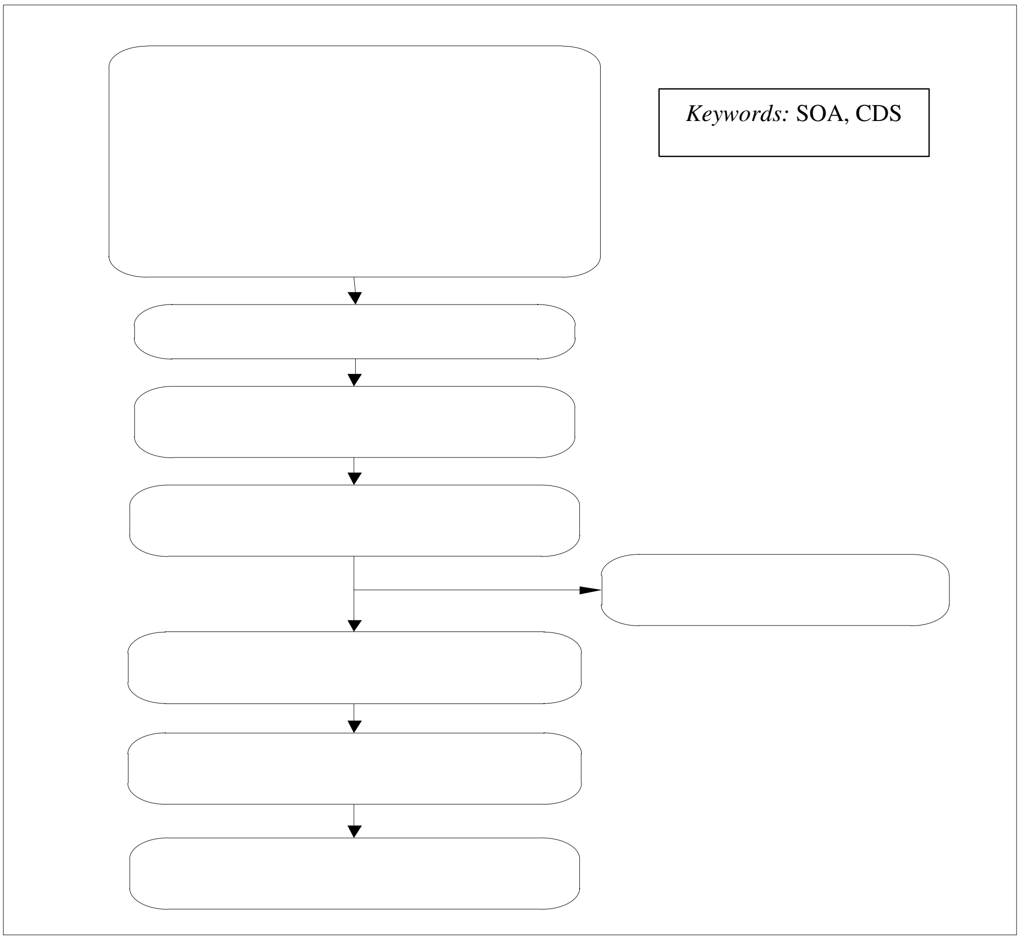


Figure 2: Study selection process.

2.2.1. Publications per year

The selected articles were classified by year of publication in order to appreciate the research activity in the area of SOA for CDS. According to this systematic review the use of SOA for CDS appeared for the first time in 2004. Figure 3 shows that maximum number of articles that were published in 2009 and that there is a slight decline in the number of publications in 2011 and 2013. It is important to note, that the systematic review only includes studies available as of October 2013.

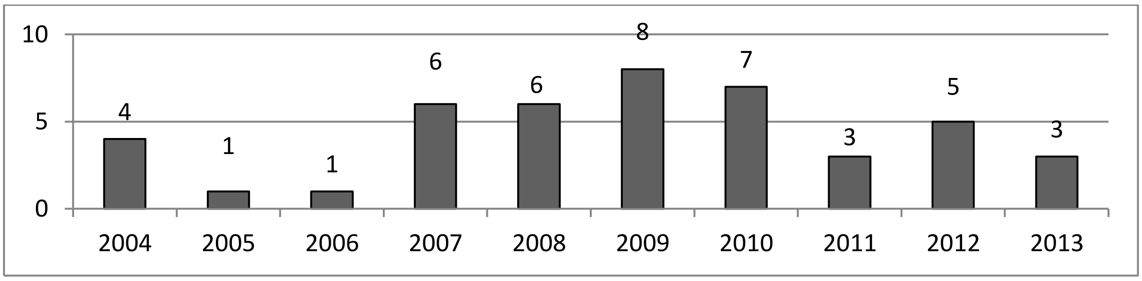


Figure 3: Number of publications per year.

2.2.2. Architectural Approach

The systematic review identified the following six architectural approaches in the use of SOA for CDS. It is important to note that some studies adopted more than one of these approaches. 7

- *Point-to-point communication.*
- *Enterprise Service Bus (ESB).*
- *Service registry.*
- *Clinical Guideline Engine.*
- *Rule-based Engine.*
- *Service Choreography and Orchestration.*

These categories were defined based on literature that described typical SOA approaches [6]–[10] and in the SOA approaches identified in this systematic review.

Figure 4 suggests that point-to-point has been the architectural approach most commonly used for the development of SOA-based CDS systems. Service choreography seems not to have been widely adopted for CDS implementations.

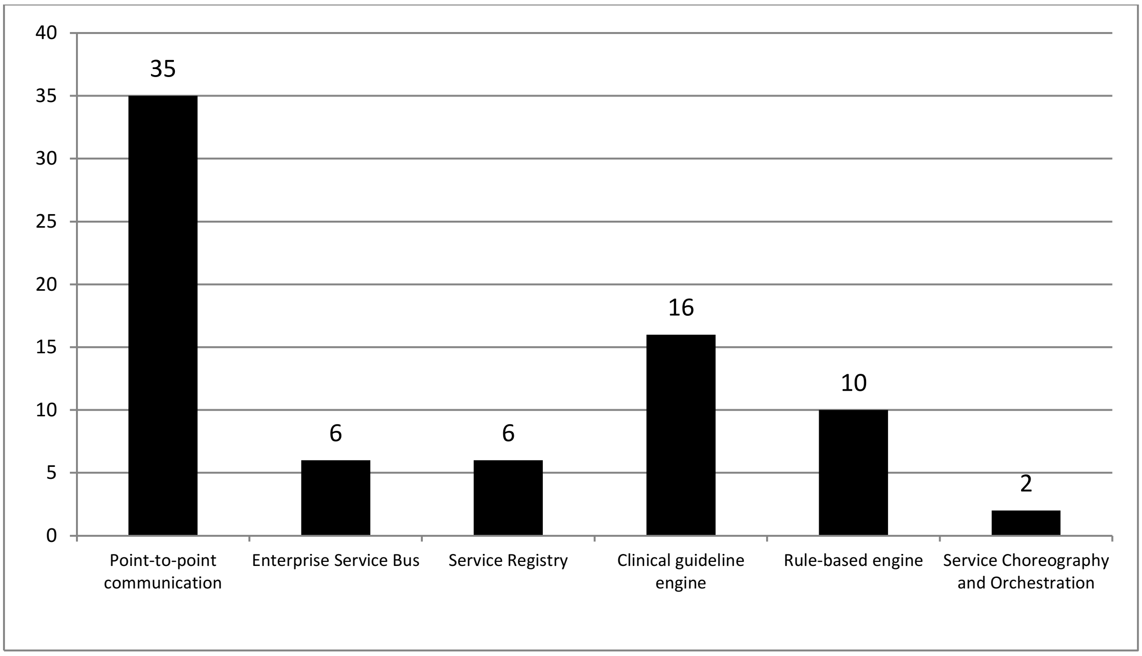


Figure 4: Number of publications per architectural approach.

2.2.3. Point-to-point communication

Most of the studies included in the systematic review adopted point-to-point communication. The articles do not provide an explanation for using this approach. However, it was assumed that this is the natural first step in evolving the integration of services. In the point-to-point communication approach, each interaction between the applications (e.g., service provider and service consumer) is individually designed, implemented, and administered. The review suggests that Simple Object Access Protocol (SOAP) is the preferred communication protocol for this approach. Table 1 shows the system name, communication protocol, and clinical implementation. One critical factor in selecting a communication technology for point-to-point communication is security, due to the need to transfer and use confidential medical data. SOAP is based on OASIS standards and has in-built security features. However, SOAP does consume more bandwidth when compared to another communication technology like Representational State Transfer (REST). Whilst REST has some advantages, it requires significant effort to make secure data communications

Table 1: Publications that adopted or suggested a point-to-point communication approach.

Communication approach	**System name**	**Clinical implementation area**	**Publication reference**
SOAP	OPNI-Web	Neonatal intensive care unit	[11]
SOAP	NewGuide	Homecare-based pressure ulcer prevention, acute ischemic stroke treatment, heart failure management	[12]
SOAP	GLEE	Childhood immunization, cough	[13]
SOAP	SAGE	Immunization	[14]
SOAP	ARTEMIS	Not specified	[15]
SOAP	--	Not specified	[16]
SOAP	--	Chronic disease management	[17]
Not specified	--	Pediatric immunization	[18]
SOAP	--	Asthma, drug-drug interaction checking	[19]
SOAP	--	Not specified	[20]
SOAP	--	Not specified	[21]
SOAP	DDSOnt	Collaborative diagnosis decision making among physicians	[22]
SOAP	EGADSS	Not specified	[23]
SOAP	EGADSS	Not specified	[24]
SOAP	SANDS	Drug interaction checking, syndromic surveillance, diagnostic decision support, inappropriate prescribing in older adults, information at point of care, personal health record	[25]
Not specified	--	Translational medicine	[26]
Not specified	--	Childhood immunization	[27]
Not specified	OncoTheraper	Pediatric oncology	[28]
Not specified	TRIACS	Diabetic retinopathy screening and research	[29]
SOAP	SAPHIRE	Hospital and homecare environment	[30], [31]
Not specified	DeGeL/Gesher	Preeclampsia/eclampsia toxemia guideline	[32]
Not specified	MATE	Cancer multidisciplinary meeting	[33]
SOAP	--	Medication management	[34]

Not specified	--	Intensive care unit	[35]
SOAP	--	Not specified	[36]
Not specified	--	Hypertension guideline	[37]
SOAP	--	Detection and automated monitoring of hospital-acquired infections	[38]
Not specified	HealthFlow	Osteoporosis	[39]
SOAP	CBPsp	Neonatal intensive care unit	[40]
SOAP	--	--	[41]
Not specified	--	Palliative care	[42]
SOAP	--	Colon cancer screening	[43]
SOAP	--	Hypertension, chronic care management	[44]
SOAP	--	Diabetes	[45]

2.2.4. Enterprise Service Bus (ESB)

The ESB is another architectural approach, which aims to provide loose coupling for applications with a universal bus that can accept all data formats from any source and integrate data flows into the appropriate applications. An ESB separates the integration logic into manageable pieces and is highly scalable [9]. There is a large spectrum of ESB systems offering different levels of functionality. Six articles that proposed or used this architectural pattern were identified. In contrast to the point-to-point communication approach, the messages pass through the ESB, which serves as an intermediary between the service provider and service consumer. Some of the advantages of using an ESB include the ability to support message routing, event triggering, data transformation, security, monitoring and management [46].

The ESB functionalities commonly used for the development of service oriented CDS were identified. The most common features are listed in Table 2. Table 2 suggests that protocol bridging and data transformation are the main features exploited, whereas only one study suggests using an ESB's event-driven functionalities to provide CDS services.

Table 2: ESB-based features used or proposed in the analysed studies.

ESB features used	**System name**	**Implementation or proposed services**	**Publication reference**
Event-driven functionalities	Infoway	Identify patterns of interactions such as spread of epidemics, distribution patterns of patients in particular regions or distribution patterns of particular health services	[47]
Protocol bridging, data transformation	HEARTFAID	Intensive care unit	[31], [48]
Protocol bridging, data transformation	COSARA	Intensive care unit	[49], [50]
Protocol bridging, data transformation	SOCBeS	Chronic disease prevention	[51]

Protocol bridging , data transformation	SCP	Multiple morbidities	[52]

2.2.5. Service Registry

The service registry supports different strategies in SOA-based systems such as standardization of service contracts, metadata centralization and notification of service contract changes to consumers [7]. The service registry is a system component that stores information related to each service (e.g., description, policies, contract location, and versions). Thus, service consumers can find the services that fulfill their requirements by querying the service registry.

Table 3 suggests that the provision of service descriptions is the main feature exploited from the service registry.

Table 3: Architectures that include a service registry.

System name	**Service registry feature used**	**Publication reference**
DDSOnt	Provide service descriptions	[22]
SAPHIRE	Store ontologies, advertise and discover Web services	[30], [31]
Infoway	Provide service descriptions	[47]
--	Provide service descriptions	[37]
SANDS	Provide service descriptions	[25]

2.2.6. Clinical Guideline engine and Rule engine

In addition to point-to-point communication, ESB and service registry, a fourth important architectural approach in the context of SOA-enabled CDS is the use of a clinical guideline engine or a rule engine. In this systematic review clinical guideline engine is considered to be a program capable of interpreting clinical knowledge expressed in a computerized format [53], whereas a rule engine is considered to be a software system that is designed to manage and enforce business rules expressed in a specified format such as if-then formats [54].

Table 4 presents an overview of which clinical guideline engines and rule engines were employed, as well as the guideline or rule languages used by these systems.

Table 4: Systems that suggest or adopt guideline engine or rule engine.

System name	Rule /guideline engine	Language	Publication reference
GLEE	GLEE	GuideLine Interchange Format (GLIF)	[13]
SAPHIRE	GLEE	GLIF	[30]
NewGuide	--	Flow-chart like approach with strong connection with Petri Nets	[12]
SAGE	SAGE	SAGE guideline model	[14]
--	ActiveBPEL	Business Process Execution Engine (BPEL)	[16]
--	Collaxa BPEL Engine	BPEL	[19]
BJC Healthcare	--	BPEL	[20]
--	SEBASTIAN	Rule-based	[17]
--	Jess	Rule-based	[19]
HEARFAID	Jena	Rule-based	[48]
EGADSS	C Language Integrated Production System (CLIPS)	Rule-based	[23], [24]
--	iLog Rules	Rule-based	[26]
--	--	Rule-based	[27]
--	SEBASTIAN	Rule-based	[34]
--	SEBASTIAN	Rule-based	[36]
--	--	Rule-based	[37]
--	Arden Syntax engine	Arden Syntax	[38]
--	--	Rule-based	[44]
SCP	OpenCDS	Rule-based	[52]
TRIACS	Triana	Triana workflow language	[29]
DeGeL/Gesher	Asbru engine	Asbru	[32]
MATE	Tallis	PROforma	[33]
COSARA	--	BPEL	[50]
--	--	SAGE guideline model	[37]
HealthFlow	Shark	XML Process Definition Language (XPDL)	[39]
--	--	BPEL	[43]

A number of standards for presenting clinical guidelines, also referred to as guideline-modeling methodologies were identified, namely GLIF, NewGuide, SAGE, Asbru, PROForma, Arden Syntax and rule-based standards *(rule-based standards can be further refined by various rule languages* – see *table 5*). It was also found that BPEL and XPDL, which are typically used for businesses other than healthcare, are also used as a representation language in these systems. The systems listed in table 4 show different trends in adopting the use of the above mentioned guideline representation standards. This is shown in figure 5.

It can be seen in figure 5 that rule-based engines are popular among the developers and that the Business Process Execution Language (BPEL) is beginning to be adopted at a higher rate than clinical guideline representation languages. Additionally, table 5 shows that there is not a clear preference for a specific rule language.

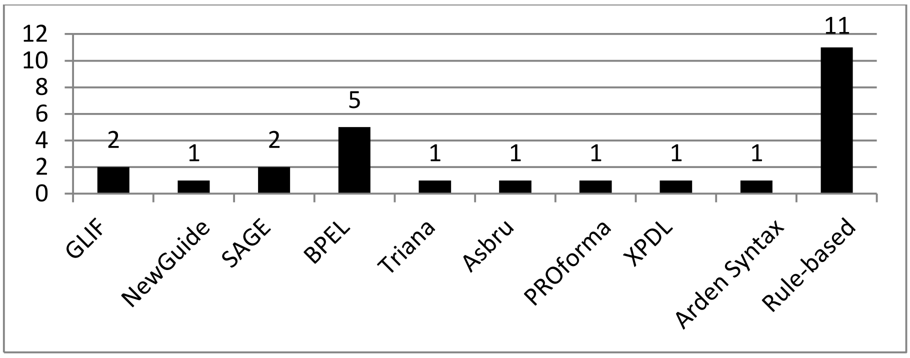


Figure 5: Number of systems per language type.

Table 5: Rule language used and publication reference.

Rule language	Publication reference
Jess	[19]
Jena	[48]
CLIPS	[23], [24]
iLog	[26]
Drools	[52]
Not specified	[27]
Java	[17], [34], [36]
Not specified	[37]
Not specified	[44]

2.2.7. Service choreography and orchestration

The final architectural approach from the reviewed studies that is evident in SOA based CDS is service choreography. In Web service choreography, each participating service defines its part in the interaction and services can interact directly with one another. Architecture closely related to Web service choreography is Web service orchestration. Unlike service choreography, in Web service orchestration (e.g., BPEL), the logic is specified by a single participant, which is referred to as the orchestrator [10]. Web service choreographies describe the observable interactions between services from a global perspective, and none of the participants controls the interaction. The survey suggests that choreography was only adopted by one research project (OpenKnowledge) [55], [56].

It can be inferred from the studies that all the architectural approaches described above can exist in multiple combinations within a SOA based CDS implementation. For example, in a setting where several healthcare organizations or units have to interact, service choreography can be used to describe the message-based interactions from a global point of view and service orchestration can be used to control the internal processes of each organization. Figure 6 describes a hypothetical scenario where all of these approaches can coexist among organizations A, B, and C. The SOAP-based interactions between the organizations are defined by the Web Service Choreography Description Language (WS-CDL) [57]. Organization C uses an orchestrator engine to deploy processes that define clinical pathways. These clinical pathways are developed based on clinical guidelines and according to the resources available in the organization. The ESB provides connectivity with other systems and monitors service interactions in order to detect health-specific patterns. Additionally, clinical knowledge is captured in the form of rules in the rule engine. The rules have the form: IF conditions THEN conclusion. These rules are integrated in the clinical pathways as tasks on the processes deployed in the orchestration engine. The private service registry contains information on each of the services available inside the organization and also provides a subscription mechanism in order to notify service consumers when a service is modified or updated. The public service registry maintains information about the services available between communicating organizations.

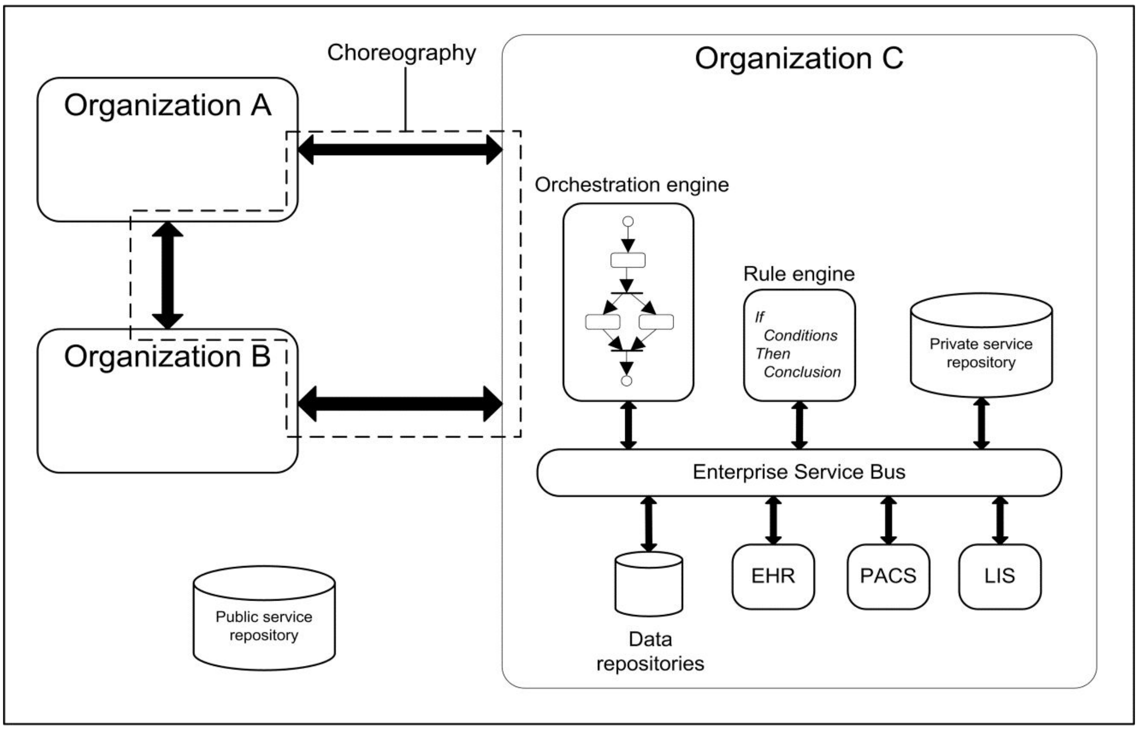


Figure 6: Example scenario: co-existence of multiple architectural patterns.

2.2.8. Healthcare standards

It is a very well established issue that semantic interoperability is essential for SOA architectures [58]. Without semantic interoperability, service providers and service consumers cannot make use of the data exchanged. For example, a service that requires past patient diagnoses using a standard terminology will not be able to properly process diagnoses provided using local proprietary codes or free text. Thus, achieving semantic interoperability is an important goal when implementing SOA for CDS.

In order to obtain an accurate picture of how semantic interoperability is accomplished, all healthcare standards adopted or suggested by the reviewed studies were extracted and categorized. The overview below describes the standard categories, and table 6 lists the individual standards identified in the specific articles. The healthcare standards identified were categorized as follows:

- *Patient information standards*: The objective of these standards is to document important patient information such as diagnoses, medications, and lab results. The HL7 Virtual Medical Record (vMR) was especially designed to integrate patient information with CDS systems [59].

- *Medical terminology standards*: These standards define a common terminology and vocabulary to be used in a healthcare domain. Some of these standards, such as the International Classification of Diseases (ICD) for diagnoses and LOINC for laboratory results, have been adopted internationally, whereas other standards, such as drug descriptions, have been adopted in specific regions (e.g., RxNorm in the United States and Anatomical Therapeutic Chemical (ATC) in Europe).
- *Standards for Web services and exchange of clinical documents*: Standards used for exchange of clinical information were grouped in this category. The HSSP Web service standards have been developed specifically to support SOA architectures. The HSSP project is a joint activity between HL7 and OMG [60].
- *Standards for medical devices*: These standards focus on the communication between medical devices and external systems.
- *Clinical guideline representation standards*: Most of these standards have been influenced by workflows and are based on XML syntax. Some of them allow the creation of XML-based clinical guidelines using a graphical editor. Comprehensive comparative reviews of some of these standards have been published elsewhere [53], [61], [62].

In summary, a large number of organizations have taken initiatives to develop and promote standards. This finding indicates a relative consensus on the need for organizations implementing SOA-based CDS to adhere to relevant standards in order to achieve semantic interoperability.

Table 6: Healthcare standards referenced by included studies.

Focus	Name	Publication reference
Clinical guideline representation	GLIF (Guideline Interchange Format)	[13], [30], [47]
	NewGuide	[12]
	SAGE (Standards-based Sharable Active Guideline Environment)	[14], [37]
	Asbru	[32]
	PROforma	[33]
	Arden Syntax	[23], [24], [38]
	GELLO (Guideline Element Language, Object Oriented)	[16]
Patient information	vMR (Virtual Medical Record)	[12]–[14], [16], [20], [25], [41], [44], [45], [52]
	CCD (Continuity of Care Document)	[44], [52]
	OpenEHR	[15], [35], [45]
	CDA (Clinical Document Architecture)	[15], [19], [23], [24], [45], [47]
	CEN EN 13606-1	[15], [30], [45]
	HL7 v2.x and HL7 v3	[25], [35], [44], [52]

Web services and exchange of clinical documents	HSSP Decision Support Service standard	[17], [21], [34], [36], [41], [52]
	HSSP Retrieve, Locate, and Update Service standard	[21], [36], [41]
	HSSP Common Terminology Service standard	[21], [36], [41]
	HSSP Identity Cross-Reference Service standard	[41]
	HSSP Healthcare and Community Services Provider Directory Service standard	[41]
	Healthcare Information Technology Standards Panel (HITSP) SC109	[41]
	IHE XDS (Integrating the Healthcare Enterprise, Cross-enterprise Document Sharing) profile	[30]
	NHIN (Nationwide Health Information Network) standards	[25]
	IHE Query for Existing Data profile	[41]
	IHE Request for Clinical Guidance profile	[41]
Medical terminology	SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms)	[14], [15], [41], [44], [45]
	LOINC (Logical Observation Identifiers Names)	[12], [14], [15], [41], [44]
	ICD (International Classification of Diseases)	[12], [19], [41]
	CPT (Current Procedural Terminology)	[41]
	UMLS (Unified Medical Language System)	[41]
	HL7	[41]
	RxNorm	[41], [44]
Medical devices	ISO/IEEE 11073-10101 (Health informatics - point-of-care medical device communication - part 10101: nomenclature)	[30]

2.2.9. Challenges and lessons learned

In addition to the architectural and standards analysis, reported challenges and lessons learned during the development and implementation of SOA-based CDS systems were analyzed. It is important to note, however, that many of the reviewed articles described planned future architectures and could not comment on actual deployment.

Lobach *et al.* reported an excessive time for data retrieval from the data repository, which slows down the CDS service considerably. They suggested introducing performance-enhancing strategies including multi-threaded data retrieval and pre-caching of patient data [17].

Wright and Sittig highlighted the problem of duplication and conflicting data distributed in different systems [25]. They solved data inconsistencies, such as different values on vital signals, height, and weight, by using only the most recent observation. Other data points, such as medication lists, were semi-automatically reconciled.

Cucino and Eccher pointed out the lack of mechanisms for defining interactions with people within the BPEL process language [43]. They suggested using BPEL4People [63] in their future work in order to overcome this limitation.

The authors of the HealthFlow system highlighted the difficulty in representing complex algorithms in graphical-based representations, such as the XPDL language [39]. The authors proposed two strategies to alleviate this problem and to simplify complex process flowcharts. First, they suggested adopting a hierarchical arrangement of scenarios where a node in a higher lever flowchart expands into a sub-flow, which may consist of multiple steps. Secondly, the authors suggested moving some of the logic into single rule-based nodes, which call a comprehensive rule base or other expert system.

Paterno *et al.* reported that the main limitation of SOA-based CDS systems evaluated in their article is the long time required to receive a CDS response [44]. For future work, the authors suggest optimizing dependent services to speed up service response times and monitoring processes across hardware and software platforms to identify and study latency issues between services.

In summary, the reviewed articles reported the typical challenges of distributed applications. However, time of response seems to be the most critical aspect for SOA-based CDS systems. Some high-risk clinical scenarios have little tolerance for latency.

2.3. Discussion

One of the main goals of the review was to identify the technologies and architectural approaches adopted for the development of SOA based CDS environments. Forty-four studies suggested or implemented service oriented CDS. The first studies in this area were published in 2004, with the most publications published in 2009. The aspects of the findings are discussed below.

2.3.1. Software architecture and development techniques

Point-to-point communication appears to be the current architectural approach of choice, whereas service choreography has been explored in only one project. A comparison of service choreography and service orchestration approaches revealed important results. In choreography, the services interact directly with one another, resulting in less overhead and delay. Using the orchestration approach, the interaction is controlled by one of the participant systems, thus increasing the time for communication [64]. It is important to understand this issue, as the speed of CDS result delivery has been identified as a key aspect in successfully implementing CDS in the clinical workflow [44]. However, orchestration allows simple Web services to perform their function without knowing they are participating in a high-level functionality of the application.

ESB is the architectural approach that is mainly used to integrate disparate applications, but only one study in our reviewed set [47] identified the event-driven features of ESB as a mechanism to provide CDS services. CDS should also exploit other features provided by the ESB, such as load balancing, service version selection, service selection based on message data, access control, and exception handling [7].

Similarly, the service registry has not been fully exploited. The main functionality of the service registry is to provide service descriptions, but it can also provide other useful functionalities, such as dependency management, event notification, access control, policy management, and federation [7].

Interestingly, none of the studies reported or suggested the use of Service Component Architecture (SCA), which is a set of OASIS [65] specifications specially designed to build distributed applications based on SOA. The SCA effort was started in 2005 by a group of vendors that includes IBM, Oracle, SAP, and others and handed over to OASIS in 2007. SCA represents the next step in the evolution of SOA, raising the level of abstraction and addressing two issues in software development: complexity and reuse [66]. Additionally, SCA hides the complexity of specifying security, reliability and other quality of service from the application code. Some of the open source SCA implementations include Apache Tuscany [67], Fabric3 [68], FraSCAti [69] and Red Hat SwitchYard [70]. The use of SCA would facilitate developers of CDS solutions to follow SOA principles and best practices.

2.3.2. Business Process languages and Web services specifications

Although there are several executable clinical guideline standards that are specific to the medical field, it appears that general-purpose business process languages, such as BPEL, are being used more often for executing simple clinical guideline logic. Interestingly, BPMN 2.0 [71], which has become the preferred standard for business process modeling

[72], has not been used by any of the reviewed systems. Unlike the BPEL language, BPMN supports human tasks, which was one of the reported challenges [43].

An essential aspect of SOA is semantic interoperability, without which service providers and consumers cannot make use of the data exchanged [58]. Efforts such as the HSSP project that provide Web service specifications that can be used to implement CDS represent a major step forward for achieving SOA for CDS. Examples of such HSSP services are the HSSP Decision Support Service, the HSSP Common Terminology Service, and the HSSP Retrieve, Locate, and Update Service.

2.3.3. HL7 Initiatives supporting SOA use in CDS

Providing common building blocks across various software vendors would greatly enhance current CDS capabilities of healthcare systems. Such common building blocks would enable greater use of SOA principles in CDS development. HL7 has a working group devoted to facilitating interoperable CDS that in the past helped developed several CDS standards, such as the Arden Syntax. In 2012, this workgroup identified 10 services and 8 capabilities as being desired for clinical information systems (CIS) to offer SOA-based CDS capabilities [41]. Examples of services identified as being desired are event subscription and notification, cohort identification and entity identification services; table 7 provides the complete list. As for the CDS capabilities identified as being desired, examples include use of standard information models and terminologies, the ability to leverage a DSS and the ability to leverage a unit conversion; table 8 provides the complete list. It is unlikely that current systems can support all of these capabilities and services. In both table 7 and table 8, include the SOA architectural approach identified in this review as an additional column that could enable this capability or service. These services and capabilities could be implemented combining SOA strategies such as ESB and business process management (BPM). The workgroup description of services with SOA patterns that could fulfill these requirements are extended.

Table 7: Services required from CIS to enable SOA for CDS identified by HL7 CDS working group (first and second columns [57]) and SOA architectural approaches that can fulfil these requirements (third column).

Service or capability	Description, example use case, and relevant standards	SOA architectural approach that could enable this capability or service
Event subscription and notification service	**Description**: Publishes relevant CIS events, which can be listened to by a CDS system to trigger CDS. Allows systems to subscribe to specific types of event notifications. **Use case**: EHR system publishes events such as the entry of new labs into the clinical data repository, patients checking into appointments, and users logging into the system. CDS system subscribes to types of events that will trigger specific CDS processes. **Relevant standards**: CORBA event service and notification service; WS-Notification	**ESB**: The ESB facilitates integration of legacy systems and can provide content/topic-based routing capabilities which can be based on standards such as WS-Notification or WS-Eventing [73]. **Examples of commercial and open source ESBs that support these standards**: *WS-Notification*: Apache ServiceMix, IBM WebSphere, SwitchYard ESB *WS-Eventing*: WSO2 ESB
Cohort identification service	**Description**: Identifies a patient cohort (i.e., population) matching search criteria. The result returned is typically a list of identifiers for matching patients. **Use case**: A population health management system identifies patients with diabetes, hypertension, and congestive heart failure using an EHR system's cohort identification service **Relevant standards**: None identified	**ESB**: The ESB provides adapters for several communication protocols and mechanisms to access databases. If the CIS does not support Web services, the ESB can provide this capability or directly query databases and return the list of identifiers for the matching patients.
Entity identification service	**Description**: Identifies whether there is an individual patient matching demographic search criteria (e.g., name, gender, date of birth). Also may be applied to identify other entities such as healthcare providers or facilities. **Use case**: A vaccine forecasting system identifies whether care organizations A, B, or C have data on a patient for whom a vaccine forecast has been requested. **Relevant standards**: HSSP Identity Cross-Reference Service standard	**ESB**: The HSSP identity Cross-Reference Service standard is based on XML, which is highly supported by the ESB. Therefore, legacy systems can provide this service through the ESB.
Clinical data query service	**Description**: Retrieves existing clinical data from clinical information system. **Use case**: Drug-drug interaction alert system retrieves patient medications from an EHR system. Relevant **Relevant standards**: HSSP Retrieve, Locate, Update Service standard. IHE Query for Existing Data profile.	**ESB**: These standards are based on XML which is supported by the ESB. Therefore, legacy systems can provide this service through the ESB.

Resource query service	**Description**: Retrieves data about local availability of material and human resources. **Use case**: Based on local availability of dialysis machine, refer patient to external dialysis facility. **Relevant standards**: HSSP Healthcare and Community Services Provider Directory Service standard; Wf-XML specification (http://www.wfmc.org/wfmc-wf-xml.html)	**ESB**: These standards are based on XML, which is supported by the ESB. Therefore, legacy systems can provide this service through the ESB.
Data acquisition service	**Description**: Retrieves data directly from users. **Use case**: CDS system asks user if patient is pregnant. **Relevant standards**: HSSP Retrieve, Locate, Update Service standard. Description:	**ESB**: These standards are based on XML, which is highly supported by the ESB. Therefore, legacy systems can provide this service through the ESB.
Data addition/update service	**Description**: Updates or adds data into a clinical information system. **Use case**: Health maintenance module updates EHR that patient had an influenza vaccine at grocery store on date X. **Relevant standards**: HSSP Retrieve, Locate, Update Service standard. Description:	**ESB**: These standards are based on XML, which is supported by the ESB. Therefore, legacy systems can provide this service through the ESB.
Order placement service	**Description**: Places a clinical order. Use case: CPOE CDS module places a pending order for lisinopril 5mg PO QD. **Relevant standards**: None identified	**ESB**: The ESB provides adapters for several communication protocols and mechanisms to access databases. If the CIS does not support Web services, the ESB can provide this capability or directly query databases and return the list of identifiers for the matching patients.
User communication service	**Description**: Communicates CDS results with appropriate end users. **Use case**: A CDS system places a note in the EHR system's alert inbox; CDS system provides a popup alert; physician is paged regarding urgent CDS finding. **Relevant standards**: None identified	**ESB**: The ESB provides adapters for several communication protocols and mechanisms to access databases. If the CIS does not support Web services, the ESB can provide this capability or directly query databases and return the list of identifiers for the matching patients.
Task management service	**Description**: Allows tasks to be added, tracked, and retrieved. **Use case**: A population health management system is able to distribute the tasks from a care plan to the task lists of various users. **Relevant standards**: Wf-XML specification (http://www.wfmc.org/wfmc-wf-xml.html).	**BPM**: This capability can be provided by a business process management systems based on workflow languages such as BPMN, which allows modeling of different sets of processes such as orchestration and collaboration and also extends the definition of human interactions[71]. **ESB**: The results of the BPM can be distributed to legacy systems through the ESB.

Table 8: Capabilities required from CIS to enable SOA for CDS identified by HL7 CDS working group (1st and 2nd columns [57]) and SOA approaches that can fulfil these requirements (3th column).

Service or capability	Description, example use Case, and relevant Standards	SOA architectural approach that could enable this capability or service
Use of appropriate, standard information models and terminologies	**Description**: Appropriate, standard information models and terminologies are used to instantiate data in the payloads of various services. **Use case**: A Clinical Data Query Service uses a standard information model to represent the data it provides. **Relevant standards**: HL7 version 2 and 3 messaging standards; HL7 Virtual Medical Record standard; IHE profiles; HITSP standards; OpenEHR templates; Detailed Clinical Models; SNOMED CT; LOINC; ICD; CPT; various others.	**ESB**: Mapping technologies such as XSLT are highly supported by ESB and can be used to map proprietary CIS terminologies and models using an external terminology server such as UMLS. **BPM**: The interaction between the CIS and the terminology service can be orchestrated using a business process management system based on business process languages such as BPMN.
Ability to leverage a DSS	**Description**: The CIS is able to use a DSS to obtain patient-specific care assessments and recommendations. **Use case**: The disease management module of an EHR system uses a DSS to obtain diabetes care recommendations based on national guidelines. **Relevant standards**: HSSP Decision Support Service standard; IHE Request for Clinical Guidance profile. Description:	**ESB**: The ESB provides adapters for several communication protocols and mechanisms to access databases. If the CIS does not support Web services, the ESB can provide this capability or directly query databases.
Ability to leverage a terminology service	**Description**: The CIS is able to use a service to fulfill terminology needs. **Use case**: A CIS uses a terminology service to convert internal laboratory codes into the LOINC codes required by a DSS. **Relevant standards:** HSSP Common Terminology Services 2 standard	**ESB**: The ESB provides adapters for several communication protocols and mechanisms to access databases. Thus, if the CIS is not able to call an external terminology service, the ESB can be used to deploy a micro flow that could perform the terminology Web service call and transform the request according to the requirements of the CDS service. **BPM**: The interaction between the CIS and the terminology service can be orchestrated using a business process

		management system based on business process languages such as BPMN.
Ability to leverage a unit conversion service	**Description**: The CIS is able to use a service to convert units. **Use case**: A CIS uses a unit conversion service to convert laboratory units used by the local CIS into the different laboratory units required by a DSS. **Relevant standards**: The Unified Code for Units of Measure (http://aurora.regenstrief.org/~ucum/ucum.html).	**ESB**: The ESB can be used to consume external services and convert the response to the data format and communication protocol supported by the CIS. **BPM**: The interaction between the CIS and the unit conversion service can be orchestrated using a business process management system based on business process languages such as BPMN.
Ability to leverage a data transformation service	**Description**: The CIS is able to use a service to render structured data into a human- readable format. **Use case**: The disease management module of an EHR system uses a data presentation service to convert an XML document representing diabetes care needs into an HTML diabetes management dashboard to be presented to a clinician. **Relevant standards**: W3C XSL Formatting Objects (http://www.w3.org/wiki/Xsl-fo)	**ESB**: The ESB can be used to consume external services and convert the response to the data format and communication protocol supported by the CIS. **BPM**: The interaction between the CIS and the data transformation service can be orchestrated using a business process management system based on business process languages such as BPMN.
Ability to leverage a data presentation service	**Description**: The CIS is able to use a service to render structured data into a human- readable format. **Use case**: The disease management module of an EHR system uses a data presentation service to convert an XML document representing diabetes care needs into an HTML diabetes management dashboard to be presented to a clinician. **Relevant standards**: W3C XSL Formatting Objects (http://www.w3.org/wiki/Xsl-fo).	**ESB**: Data model transformation is one of the main functionalities of the ESB. Thus, this transformation can be performed inside the ESB using technologies such as XSLT.
Ability to populate a data warehouse in real-time	**Description**: The CIS is able to populate an enterprise data warehouse in real-time, as opposed to nightly batches. **Use case**: A healthcare organization builds CDS functionality against the data warehouse. **Relevant standards**: None identified.	**ESB**: The CIS can perform this task using the different database communication protocols supported by the ESB. **BPM**: The interaction between the CIS and the data warehouse can be defined using a business process management system based on business process languages such as BPMN.
Maintenance of audit logs	**Description**: The CIS maintains an audit log of all service interactions. **Use case**: A CIS maintains an audit log of data provided to, and recommendations received from, an external CDS service. **Relevant standards**: Healthcare Information Technology Standards Panel (HITSP) SC109.	**ESB**: Audit logs of service interactions are often part of the monitoring and management tools provided by the ESB [7]. Additionally, other technologies such as Business Activity Monitor (BAM) and Complex Event Processing (CEP), which are commonly used on SOA architectures, can also be implemented for this purpose [74].

2.4. Limitations

The search was limited to English-language articles published until the end of October 2013. In addition, our review is based on published literature only, whereas there could be additional relevant CDS implementations that have not yet been described in the literature.

2.5. Conclusion

The great promise of SOA for CDS can be achieved more rapidly if best practices identified in other industries such as finance are adopted. Several technologies and methodologies have been developed around the concept of SOA, and they have the potential to enable a new range of opportunities for CDS. These include BPMN for optimizing clinical pathways and SCA for Web service composition. It is recommend to further explore the service choreography for use in SOA based CDS implementations.

Chapter 3

Evolution of Clinical Decision Support Systems: Health Information Technology, Standards and Implementation Approaches

3.1. Introduction

This chapter presents the background theory and concepts for this thesis. The field of health information technology (HIT) is described in detail along with the prevailing challenges to effective HIT adoption and suggested strategies to overcome these barriers taken from the literature search. The scope of this thesis is defined as: HIT in the domain of Clinical Decision Support (CDS). For implementing an architectural solution for CDS, the core constructs of enterprise architecture (EA) within the realm of software architecture are explored. The core concepts of Service Oriented architecture (SOA) along with Business Process Management for an enterprise wide implementation of CDS are reviewed. The current literature about Business Process Management (BPM) and Service Oriented Architecture (SOA), along with the research gaps in the architecture stage of the implementation is identified. Information sources such as ACM Digital Library, Compendex, IEEE, Xplore, Science Direct, Scopus, Springer, Web of Science, Google scholar, dissertations, and books were utilized to gather information. This chapter has elicited important areas of research that are relevant for consideration while designing an architectural solution for a healthcare organization. The research questions derived as a result of this survey are presented. The research findings from this chapter are considered as a basis for designing and implementing a SOA based architecture solution in the later chapters but first, the theories related to HIT and CDS are reviewed.

3.2. Healthcare Information Technology

The U.S. Health and Human Services (HHS) Office of the National Coordinator for Health IT (ONC) defines Health Information Technology (HIT) as “the application of information processing involving both computer hardware and software that deals with the storage, retrieval, sharing, and use of health care information, data, and knowledge for communication and decision making” [75]. HIT encompasses a wide range of applications such as electronic health record (EHR) systems, picture archiving and communication systems (PACS), laboratory information systems (LIS) and clinical decision support systems (CDSS). According to a systematic review of 257 studies conducted by Chaudhry et al, the use of HIT: improves quality of service (e.g., adherence to guideline based care, enhances surveillance and monitoring, and decreases medication errors) and efficiency (e.g., decrease of utilization of care) [76]. The benefits that HIT provides to the patients have been recognized by several industrialized countries, which subsidize the acquisition of HIT with public funds [77]. For example the federal government of the USA assigned $19 billion to promote the adoption and use of HIT as a part of the American Recovery and Reinvestment Act of 2009 (ARRA) [78] and the government of the UK invested £2.3bn in the national programme for information technology (NPfIT) of the National Health Service (NHS) in 2002 [79]. A major objective of these efforts is to enable different healthcare providers (e.g., specialists, hospitals, nursing homes, etc.) that are distributed around the country to share and access patient data and provide diagnosis and treatment improved by the use of IT.

3.2.1. HIT Adoption barriers

Alongside the potential benefits that HIT can provide to the healthcare industry, several research articles and agency reports have identified barriers to HIT adoption from different perspectives. The key barriers identified from these studies are reviewed below:

Resistance to Organizational Innovation:

There is an observed resistance to change from a number of stakeholders in a healthcare organization (HCO). These include medical as well as non-medical actors (physicians and HCO managers). There are concerns that quality of care, efficiency and the patient-provider relationship may be negatively affected by the implementation of information systems [80], [81]. The staff usually refuse to allow changes that result from automating well-established workflows [80], [82]. Also, there is a lack of trained personnel that could lead the implementation and adoption of HIT [83].

Financial barriers:

The high initial costs as well as maintenance costs are considered to be the main factors that prevent the implementation of HIT [84]–[86]. The purchase of an EHR system is estimated to cost $33,000 per physician, with an additional maintenance cost per month of $1500 per physician [84]. Those who invest in HIT, only see about an 11% return on investment (ROI) [83]. This places the HCO’s managerial level under substantial financial liability.

Technological Barriers:

Healthcare has a complex organization [87]. From a technical point of view, the development of information systems for a complex organization requires a significant amount of effort. A key requirement for HIT is that systems must be developed based upon established workflows. Automating these workflows entails a level of process re-engineering to modify existing workflows so that the implementation reflects the routine functions of the HCO [83]. To facilitate the different actors of a healthcare organization, the HIT implementation must cater for the differences in the requirements of these actors (e.g. nurses, physicians), clinical situations (e.g., acute and chronic care), clinical environments (e.g., intensive care and ambulatory care), and institutions [88]. HIT usability is another critical factor for adoption. Studies have shown that the more intuitive and familiar the system is to the user, the more likely it is accepted [80]. However, this depends greatly on the user's previous experience with the systems in use by the institution [80]. In addition to user requirements, Health information systems appear to be particularly difficult to debug, modify and understand by non-medical professionals [88].

Interoperability among disparate healthcare knowledge resources:

Patient information is usually distributed in isolated systems within hospitals, physician offices and pharmacies [86]. Another key challenge is the interoperability between different healthcare systems [82], [86]. Existing systems developed by different vendors usually use proprietary internal representations of clinical data, which is incompatible with other systems [84]. Medical terminology is complex and varies according to medical speciality, medical facilities and clinical context [84]. For example, the abbreviation "MS" stands for "mitral stenosis" in cardiology, "multiple sclerosis" in neurology, "morphine sulphate" in anaesthesia, and "magnesium sulphate" in neurology [84]. There are also concerns about privacy and security of patient data [80].

3.2.2. Strategies to address HIT Adoption barriers

A number of strategies have been suggested by various authors providing mechanisms to overcome the barriers described above.

Overcoming Resistance to Change

Studies have shown that resistance to change can be overcome by assigning a physician as the project leader and including different members of the staff in the implementation team [80]. Usually, the staff have a better understanding of the existing processes and can provide different perspectives and expertise to the implementation [80]. Staff training in the use of the systems as well as support from experts after the implementation are considered important factors for a successful adoption of HIT [80]. An incremental implementation approach, where the functionalities of HIT are made available to the users gradually is recommended for large organizations with complex processes [80].

Managing Financial barriers

To overcome financial barriers, countries such as United Kingdom, United States and Sweden have subsidized the purchase of HIT [85], [86]. This has increased the number of physicians that make use of HIT in their daily tasks. To address concerns about privacy and security the U.S. government has begun to regulate electronic health information [89]. This regulation establishes standards, limits and conditions on the use and disclosure of patient information.

Addressing Interoperability

Several organizations realized that, in order to address the issues of interoperability in HIT, it was necessary first to agree on the meaning of the term "interoperability" [90]. The HL7 EHR Technical Committee's (TC EHR) interoperability working group analysed over 100 definitions from different institutions and identified three types of interoperability [90]:

- *Technical Interoperability*: It focuses on the physical transition of data, not in the meaning.
- *Semantic Interoperability*: It communicates meaning and is required to implement applications such as intelligent decision support. There are several levels of semantic interoperability depending on the level of agreement on the data content terminology and the content of data models.
- *Process Interoperability*: It ensures that the information is provided in time-, event-, or a sequence-oriented manner that can be used in an actual workflow of the care team.

3.2.3. HIT Standard Development Organizations

Some Standard Development Organizations (SDOs) have been formed specifically to develop standards to support HIT. Some of the most important SDOs include: HL7, ASTM E31, DICOM, CDISC, OMG, OASIS, IHTSDO, CEN TC 251, ISO TC 215 and NCHS.

A brief description of these SDOs is presented in table 9.

Table 9: Description of some significant Standards Development Organizations

SDO	Description
HL7	HL7 [91] takes its names from the International Standard Organization (ISO) standard networking levels where the seventh level is the application level. Standards developed include: HL7 v2, HL7 v3, GELLO, Arden Syntax, CCOW (Clinical Context Object Workflow), Claims Attachments, CDA (Clinical Document Architecture), Electronic Health Record (EHR) System Functional Model and Structured Labelling.
ASTM E31	ASTM Committee E31 is part of the American Society for Testing Materials (ASTM) and focuses on the architecture, content, storage, security, confidentiality, functionality and communication of electronic healthcare information and knowledge. To date E31 has over 30 approved standards [92] including Continuity of Care Record (CCR).
DICOM	The Digital Imaging and Communications in Medicine (DICOM) standard is developed and supported by the national electrical manufacturer's association (NEMA). NEMA along with HL7 developed a standard for transmitting diagnostic images, metadata, results and patient information [93]. Standards include, Picture Archiving and Communication Systems (PACS) and interfacing with medical information systems [94].
CDISC	The Clinical Data Interchange Standards Consortium (CDISC) develops and supports data standards for medical research [95]. CDISC has developed Data Tabulation Model (SDTM) with US National Cancer Institute Enterprise Vocabulary Services (EVS) [96]
OMG	Object Management Group (OMG) develops and supports standards for embedded and specialized systems, analysis and design, middleware for industries including finance, life sciences research, manufacturing technology and healthcare [97].
OASIS	The Organization for the Advancement of Structured Information Standards (OASIS) drives the development, convergence and adoption of open standards for the global information society. Standards for security, cloud computing, SOA, web services, electronic publishing, emergency management and healthcare. [98]. Some of the OASIS standards adopted by healthcare includes SAML [99], and WS-Trust [100].
IHTSDO	The International Health Terminology Standards Development Organization (IHTSDO) maintains health terminologies and standards like SNOMED CT [101]. SNOMED CT is one of the most comprehensive, multilingual terminologies used by HIT in the world [102]
CEN TC 251	The European Committee for Standardization (CEN) is the main provider of European Standard and technical specifications [103]. CEN TC 251 develops health informatics standardization [104].

ISO TC 215	The ISO TC 215 develops standards for health Information and Communication Technology (ICT) to promote interoperability between independent systems and measures to reduce duplication of effort and redundancies [105].
NCHS	The National Center for Health Statistics (NCHS) provides statistical information that can guide actions and policies to improve the Health of the American people [106]. The NCHS established the Public Health Data Standards Consortium (PHDSC).

HIT Implementation Strategy:

For implementing HIT in a healthcare enterprise, nationwide HIT initiatives have developed maturity models that focus on the management of the implementation of HIT. Some of the most important maturity models include the Electronic Healthcare Maturity Model (eHMM) developed by Quintegra [107], the Electronic Health Record Adoption Model (EHRAM) established by the Healthcare Information and Management Systems Society (HIMSS) [108], [109] and the Electronic Patient Record (EPR) maturity model established by the United Kingdom National Health Service (NHS) [110].

3.2.4. CDS as a key enabler of HIT

There are many aspects of HIT implementation like Computerised Physician Order Entry systems (CPOE), Laboratory Information systems (LIS) and Radiology Information systems (RIS) and pharmacy systems. The US Meaningful use criteria [111] identifies a number of IT capabilities that are termed as standard requirements for implementing HIT in an HCO. For an IT solution to acquire Meaningful Use certification, it has to comply with the three stages of the criteria. The third stage of this criterion is enabling Clinical Decision Support (CDS) by the use of IT to improve patient outcome. In addition to Meaningful Use criteria, the maturity models defined in the previous section identify CDS as an essential aspect of achieving advanced HIT capabilities. The Institute of Medicine (IOM) has also recognized CDS as an important strategy for improving the quality of patient care [112]. It has been observed that the aforementioned barriers to HIT adoption have also influenced the adoption of this particular functionality within the healthcare service delivery cycle. Since the focus of this thesis is on implementing an architecture solution enabling clinical decision support at the point of care, the remaining sections will present an overview of a description of CDS core functionalities and the prevalent architectural approaches for enabling CDS within a healthcare IT system. Section 2.3.1 onwards discusses CDS in detail, in-line with the organizational influences in its adoption.

3.3. Clinical Decision Support

Clinical decision support (CDS) provides clinicians, patients, or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care [2]. CDS is an important component of HIT as it has the potential to support decision making, improve quality of service, reduce costs and increase efficiency [3]. The contributions that result from incorporating CDS in HIT have been recognized by a number of maturity models that measure the implementation levels of HIT. The Electronic Healthcare Maturity Model (eHMM) developed by Quintegra [107], the Electronic Health Record Adoption Model (EHRAM) established by the Healthcare Information and Management Systems Society (HIMSS) [108], [109] and the Electronic Patient Record (EPR) maturity model established by the United Kingdom National Health Service (NHS) [110]; all identify CDS as an essential aspect of achieving advanced HIT capabilities. Similarly, the Institute of Medicine (IOM) recognized CDS as an important strategy for improving the quality of patient care [112].

2.3.1 Classification of Clinical Decision Support Systems according to functionality

CDS systems are designed to aid in the clinical decision-making at the point and time when these decisions are made. Clinical decisions encompass a wide range of activities including the classic diagnostic (e.g., determine the causes of the patient's symptoms), diagnostic process (e.g., decide what questions to ask, decide which procedures to perform), and the management of decisions (e.g., decide how to use the patient's response to therapy in order to determine an alternative approach or find out whether the initial diagnosis was incorrect) [113]. CDS systems can be classified into three categories based on the type of assistance provided: a) retrieve online documents relevant to the current clinical scenario, b) provide alerts, reminders or other information specific to the patient and clinical situation, c) organize and present information in a form that emphasizes relevant information and facilitates decision making, for example dashboards and graphical displays [113].

Retrieve online documents relevant to the current clinical scenario

These CDS tools support clinical decision making by educating the physician through information retrieved from online resources. This is usually achieved through context-aware links to online knowledge resources that are embedded into clinical systems. These links are known as “Infobuttons” [114]. It has been observed that the easy access to knowledge at the time it is required can potentially support clinical decision making [114]. It is important to note that this type of CDS is not focused on specific decision making, but only in providing information that may be relevant to the physician.

Provide alerts, reminders or other information specific to the patient and clinical scenario

These CDS systems provide recommendations according to patient specific information. These systems can be divided into knowledge-based and non-knowledge-based systems [113]. Knowledge-based CDS uses information that resides in a knowledge module. A knowledge module consists of concepts derived from domain specific information. For example a knowledge module for clinical decision support consists of clinical symptoms for a disease. These symptoms can be represented using business rules (*clinical rules*). This knowledge encoded in these rules can be used by a CDS to deduce a diagnosis. The diagnosis information is then provided to an end user in the form of alerts and reminders. Knowledge-based CDS, which relies on such rules, is termed as: a rule-based system. Non-knowledge based CDS function on the principles such as statistical data, heuristics and artificial neural networks [115].

Organize and present information in a form that facilitates decision making

This type of CDSs focuses on graphical presentation of knowledge. An example of such systems is Clinical Dashboards. The clinical dashboards result from the underlying clinical data and reasoning to present knowledge to the end user in a way that is readily understandable and can provide the necessary data to support decision-making in real time. For an example clinical scenario, a dashboard can present information in the form of graphs representing a set of population for whom a vaccination program needs to be conducted. The logic is derived from the underlying patient data for a healthcare facility. This data includes demographic details and past vaccination records for the patients registered at this healthcare facility. Dashboards are also a useful tool to analyse trends in certain medical conditions for a number of patients.

3.3.1. Implementation approaches for Clinical Decision Support Systems

Incorporating CDS in an HIT is a critical task, which requires a significant amount of planning from both the medical as well as non-medical members of the HCO. An important factor when designing the architecture for a HIT is identifying how CDS can be beneficially realized within that architecture. Clinical decision support systems (CDSS) can be classified according to the architectural approach for implementation and the capabilities of the system. For instance, Wright and Sittig categorized CDSS into four groups based on the architectural approach used for implementation [116]:

1. **Stand-alone**: The CDSS does not interact with any other system; therefore, the user has to introduce patient information in order to get guidance.
2. **Integrated**: The CDSS is strongly coupled with other healthcare systems such as computerized physician order entry (CPOE) system. One disadvantage of this approach is that sharing CDS knowledge with other systems becomes problematic.
3. **Standard-based**: Use of standards to represent, store and share decision content.
4. **Service-based**: The CDS is offered as a remote service. This loose coupling allows sharing CDS with several systems and separately maintained CDS content.

It is important to note that CDS implementations may adopt more than one of these approaches. Some researchers have demonstrated that “standard-based” and “service-based” are a successful combination to deliver CDS at the point of care [23], [25], [36]. CDS delivers better results when it is integrated into the clinical workflow and delivered at the point of care [117] [118]. An example of this approach would be an alert that is displayed on the EHR system, informing about a drug-drug interaction when the patient record is being modified.

Fox et al., presented a classification based on the capabilities of different CDSS and provide examples of implementation techniques (table 10) [119].

Table 10: Classification of CDSS according to Fox et al [50]

Capabilities	**Example Implementation techniques**
Monitoring, alerts and reminders	Algorithmic and rule based methods
Focusing and information retrieval	Search engines, navigation, infobuttons

Framing and making decisions	Decision analysis, logical decision models
Support for complex and multidisciplinary care	Workflow

Providing CDS as a part of Clinical Workflows

It has been observed that most of the CDSS limit their functionality to deliver alerts and reminders, information search, retrieval and filtering services. CDS if integrated as a part of clinical workflow can significantly improve the patient outcome [3]. This capability is still in its infancy as there is a lack of support for automated workflow management for care planning [119]. Care management using workflows takes into account the longitudinal care of a patient starting from the initial encounter, the duration of the encounter, emergency encounters, intermittent tasks like therapies and follow-up procedures. [120]. This necessitates the integration of a CDS capability in the clinical workflow in order to maximize the benefits of swift decision making by the physician.

According to [119] CDSS developers often focus on providing solutions for an individual point of care rather than the process of care. CDS can be integrated as a part of the care process.

3.3.2. CDS Adoption challenges

Despite the benefits demonstrated by various trials and reviews, the adoption rate of CDSS remains low [2], [121]. Some of the reasons include technical factors such as complexity to share CDS knowledge with other systems [118]. Figure 7 shows ten challenges for CDS adoption identified by Sittig et al., based on their practical experience [122]. The authors grouped these challenges into three categories: 1) Improve the effective of CDS interventions, 2) Create new CDS interventions, and 3) Disseminate existing CDS knowledge modules.

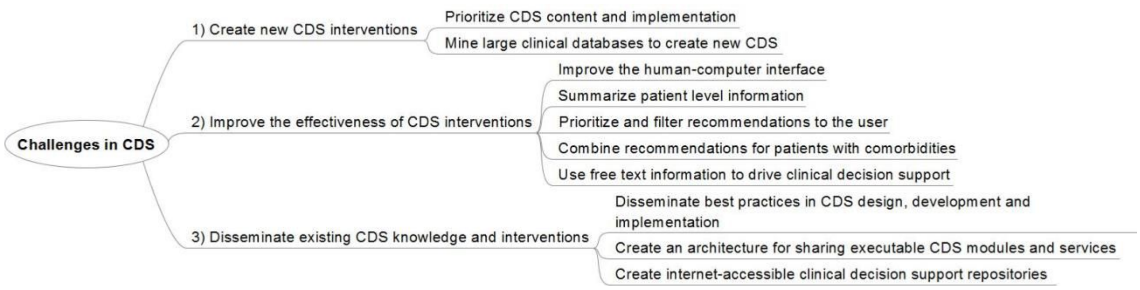


Figure 7: Challenges in CDS identified by Sittig et al., [122]

The third category illustrates the need for creating an architectural approach for sharing executable CDS modules and services. This caters for the need to maintain an independent CDS implementation, which is separate from the organizational healthcare infrastructure and can be consumed or reused by several healthcare organizations with minimal effort. Sittig et al., suggested two options to achieve this goal: 1) design knowledge modules that can be loaded into a clinical information system, or 2) execute CDS as a remote service using a standard interface through the network.

Fox et al., [119] continued the discussion about the challenges defined by Setting et al., and highlighted the following four research areas that formally address CDS related issues:

- **Decision-making and decision theory**: It is related to research on reasoning, problem solving, and decision-making. Disciplines that investigate these topics include applied mathematics, statistics, economics, and computer science.
- **Process modelling and workflow**: Examples in this area include development of formal notations for modelling and automating business process such as Petri Nets [123].
- **Knowledge representation**: It is related to research on clinical knowledge representation. Some efforts in this area include the development of clinical computer-interoperable guideline modelling languages [53].
- **Organizational theory**: Investigates the complexities of human organizations and how the introduction of technology can assist an organization achieve its goals.

From the above four research areas, “Knowledge Representation” and “decision –making and decision theory” are important for CDS features like: summarizing patient information, prioritizing and filtering recommendations and improving computer user interfaces.

From an implementation point of view, Fox et al. [119] identified the following four challenges in CDS:

- **Joint and distributed execution of plans within organizations**: It is related to the complexity in executing care plans. Care plans usually consist of many services provided by different teams, these services could be a part of multiple workflows or sub-plans.
- **Management of patient over time**: Challenge of dealing with clinical pathways for longitudinal care.
- **Joint and distributed decision-making**: This is about how to support group decision making. For example for a patient in oncology, the care is provided by a number of actors from different departments. Surgical teams for biopsies, Laboratory for test results, Specialist oncologist for care and treatment plan, and specialized medication department like chemotherapy. This scenario requires actors from these different teams to provide timely care to the patient in order to prevent worsening of the condition. A CDS capability allowing collaborative management of such a clinical scenario would result in well-timed diagnosis and treatment of the patient.
- **Managing distributed knowledge and data within and between organizations**: This is related to the complexity to manage data that is distributed among multiple sources (e.g., medical record, medical devices, and healthcare systems).

Addressing these challenges requires a holistic approach that integrates patient information distributed throughout the organization. The following concept map (figure 8) shows these challenges (round boxes) and the research areas (ovals) that address them.

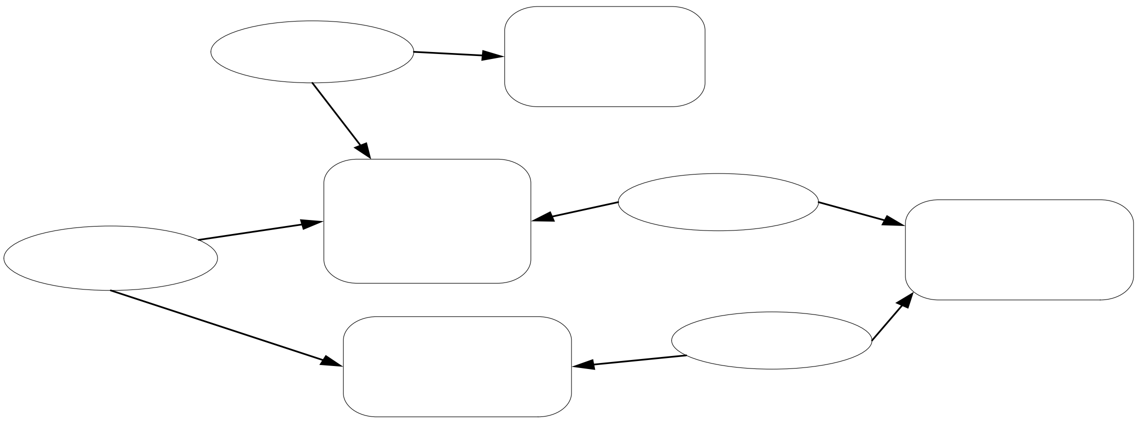


Figure 8: Challenges in CDS identified by Fox et al., [119].

In addition to the challenges reported by Sittig et al., [122] and Fox et al., [119] the HL7 CDS work group identified a set of capabilities and services (figure 9) that health information systems (HIS) must support in order to facilitate a service-oriented implementation of CDS [41]. It is highly unlikely that all of these functionalities are supported by the current systems. Therefore, another concern that arises is how to provide these capabilities to the systems currently available in the organization.

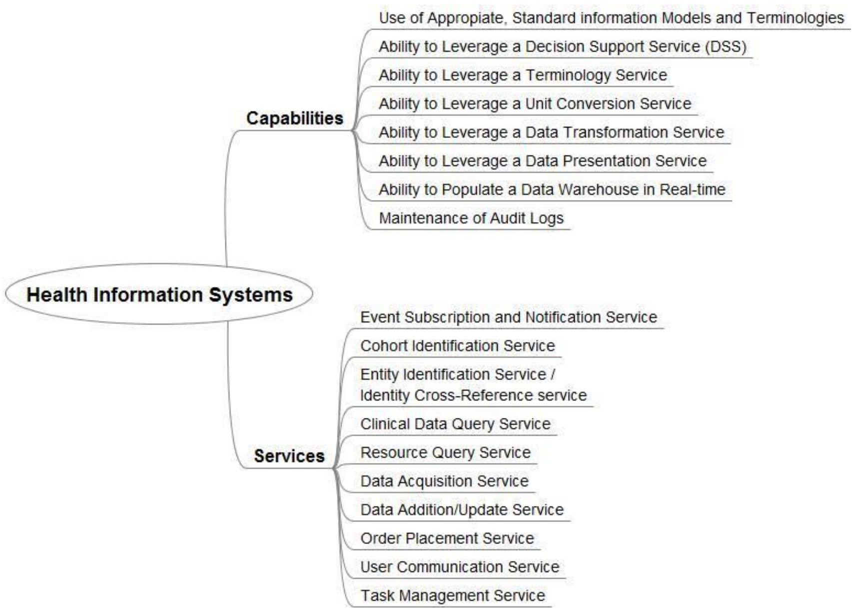


Figure 9: Service and capabilities required from HIS to enable a service-oriented CDS, adapted from [41].

Based on this analysis, it can be speculated that in order to address many of the challenges in CDS, an information technology (IT) architectural approach that incorporates the key strategic goals of the health care organization is required. Adopting a standards based approach for CDS in HIT will result in long-term benefits for the organization in terms of addressing interoperability between legacy and new systems. The next section describes the standards currently being used for enabling standards-based CDS in a HIT.

3.4. CDS related standards

Arden Syntax

Arden Syntax is a standard maintained by HL7 that allows representation of clinical knowledge in an executable format [124]. Arden Syntax was specifically designed to implement CDS systems. The standard provides mechanisms to specify variables that contain health information and to define clinical decision support logic in a language that it is easy to understand by domain experts. In Arden Syntax the CDS logic is independent from the implementation details (e.g., programming language of the CDS system). The CDS logic is organized in self-contained files known as Medical Logic Modules (MLMs), which can be executed by direct invocation or by events. The main weakness of this standard is the lack of support for the semantics of the data that is used by the MLMs [113], [124], [125] . This problem is commonly known as the "curly braces problem" [124]. The expressions inside the curly braces define the interaction between the MLMs and the healthcare system. The curly brace expressions are highly coupled to the specific implementation of the CDS, which has prevented the creation of a repository of MLMs and knowledge bases that could be shared between different institutions [124]. A proposed solution to the curly braces problem is the integration of the expression language GELLO [126] into the MLMs and use of the standard Virtual Medical Record (vMR) [59] to define an interface between the healthcare information system and the CDS system [124]. GELLO provides a standardized query language to access data from healthcare information systems and the vMR allows interfacing proprietary database models and standard-based CDS systems. Figure 10 shows how this approach would allow developers to connect standards-based CDS systems to any vMR-complaint healthcare system.

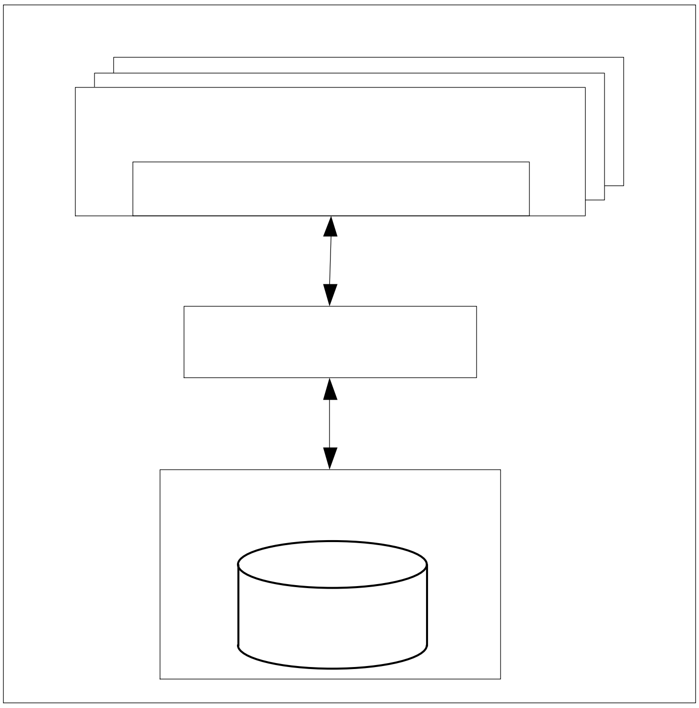


Figure 10: Proposed solution for the "curly brace problem", adapted from [113].

Virtual Medical Record (vMR)

One of the main barriers that prevents the implementation of CDS at scale is the wide variety of information models, terminologies and knowledge bases used by health information systems [2]. Without a common clinical information model, implementation, maintenance and transferring of data between CDS systems and other healthcare systems becomes problematic. The Virtual Medical Record (vMR) is a standard information model that was designed to solve this problem [127]. It includes a set of clinical data elements required to implement CDS and defines the inputs and outputs between CDS systems and health information systems.

Decision Support Service Specification (DSS)

Several organizations and governments have recognized the benefits that Service Oriented Architecture (SOA) can provide to health care and have created a global community known as Healthcare Services Specification Project (HSSP). SOA is a computing paradigm for organizing and consuming distributed functionalities, this concept is described in more detail in section 2.5. HSSP is a joint effort between Health Level 7 (HL7) and the Object Management Group (OMG) that focuses on service specification standards based on SOA principles [60]. HL7 is an international authority on standards for interoperability of health information technology [128] and OMG is an international authority on computer standards for a wide range of industries [97]. The Decision Support Service (DSS) is one of the standards created by the HSSP. The DSS receives patient data as input and returns patient-specific assessments and recommendations. In order to request an evaluation to the DSS, the service client must specify which knowledge module will be used for the evaluation, and also submits the patient data required for the knowledge module. Conceptually the DSS plays the role of a gatekeeper of one or more modules of medical knowledge. Every module of clinical knowledge is capable of using patient data in order to provide machine-interpretable conclusions. The DSS can accept as payload patient data representations based on standards such as Consolidated Clinical Document Architecture (C-CDA) or vMR [129]. The first release of the DSS specification only supported SOAP Web services [129]. The first open source implementation of this standard has been developed by OpenCDS, which is a multi-institutional collaborative effort lead by Dr. Kensaku Kawamoto [130]. At the moment of writing this thesis, OpenCDS only supports payloads based on vMR as the information model standard.

Retrieve, Locate, and Update Service Specification (RLUS)

Retrieve, Locate, and Update Service (RLUS) is another Web service standard result from the HSSP effort. RLUS defines a set of interfaces that allows locating, accessing and updating health data from healthcare organizations [131]. RLUS introduces the concept of "semantic signifier," which facilitates use of the functionalities of the service with different data models. For example an RLUS service can support payloads based on standards such as vMR or CDA.

3.5. Importance of Software Architecture

The International Organization for Standardization (ISO) defines architecture in the context of software engineering as "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" [132]. The Open Group Architecture Framework (TOGAF) standard provides two definitions for architecture, depending on the context [133]:

a) "A formal description of a system, or a detailed plan of the system at a component level to guide its implementation"
b) "The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time".

3.5.1. Enterprise Architecture Framework

EA is defined by the MIT Center for Information System Research as "the organization for business process and IT capabilities reflecting the integration and standardization requirements of the firm's operation model" [134]. In order to develop complex IT solutions, industries usually adopt an Enterprise Architecture Framework (EAF) that supports Enterprise Architecture (EA) [135]. The use of an EAF streamlines and simplifies architecture development, provides a comprehensive coverage of the selected solution and ensures that the designed architecture can grow according to the organization needs [136]. Some of the most important EA Frameworks include Zachman [137], TOGAF standard [138], Federal Enterprise Architecture Framework (FEAF) [139], Department of Defence Architecture Framework (DoDAF) [140], Gartner [141] and Open group ArchiMate

It is important to note here that organizations may adopt more than one EA framework. This approach is adopted at any stage during the enterprise life cycle depending upon solving a problem that pertains to the architectural layer concerned [135]. The survey conducted by Cameron and Mcmillan indicates that the TOGAF standard is commonly used as a process for building the technology layer, Zachman for taxonomy, Gartner for business architecture, FEAF for reference models and segment architecture, and DoDAF for governance [135]. Since the focus of this thesis is built on the grounds of enterprise architecture, The Open group ArchiMate specification is commonly used in conjunction with technological specifications in the TOGAF model. ArchiMate represents the interactions among different components that built the enterprise architecture; ArchiMate is discussed in detail in the next section.

3.5.2. The Open Group ArchiMate® standard

Some of the main aspects of the EA include identify and express the needs of an enterprise as well as establish goals or expectations and plan of action to achieve them. ArchiMate is an open and independent modelling standard that was developed to support these aspects of the EA [142]. This modelling standard focuses on the overview and coherence instead of the specificity and detail, as do other modelling languages such as UML (Unified Modelling Language) and BPMN [143]. An important feature of ArchiMate is that it allows the architect to accurately model, analyse, and visualize relationships between business domains [142]. ArchiMate was originally developed by a collaborative research project that involved several Dutch research institutes as well as government and financial institutions [143]. The standard was later transferred to the Open Group and evolved to be fully aligned with the TOGAF standard [142]. The various notations of the ArchiMate standard are described in detail in Appendix C.

3.6. Service Oriented Architecture

Service Oriented Architecture (SOA) is an architectural style that sees services as the functionalities that a system or business can deliver. These services can be used to build new business processes or applications.

Definition of SOA

The following are some of the definitions commonly used to describe SOA:

- Thomas Erl defines SOA as “an open, agile, extensible, federated, composable architecture comprised of autonomous, Quality of Service (QoS)-capable, vendor diverse, interoperable, discoverable, and potentially reusable services, implemented as Web services” [58].
- The OASIS Reference Model for SOA (SOA-RM) defines SOA as “a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains”[144]. Capability is then defined as “a real-world effect that a service provider is able to provide to a service consumer” [144].
- Papazoglou defines SOA as “a logical way of designing a software system to provide services to either end-user applications or other services distributed in a network through published and discoverable interfaces” [145]

Importance of SOA from an EA point of view

There are several benefits of applying SOA when developing software. The SOA-RM emphasizes that adopting the SOA paradigm enables interoperability, growth and reuse of software assets [144]. SOA also increases the return of investment (ROI) by facilitating integration of legacy systems with contemporary applications, accelerating development of new solutions and enabling rapid changes to meet the challenges of the organization [58]. These benefits have made SOA the preferred architectural style for designing enterprise applications according to a report published by Gartner [146].

SOA Infrastructure Layers

A SOA-based system usually comprises various abstract layers. Each of these layers has different functional properties and is managed at a different level of detail. Figure 11 shows an example of a SOA-based service in a manufacturing business. This model shows the following six levels [145]:

1. **Business Domain**: This layer contains services of the area of interest of the organization. Figure 11 shows a service called "distribution" which is one of the services that matters to customers. This service is composed of three business processes (e.g., purchasing, order management, and inventory).
2. **Business Processes**: This layer provides high-level business process (e.g., purchasing, order management, and inventory). These business processes are multi-step business interactions that use less complex services such as create an order, schedule an order or check order progress.
3. **Business Services**: This layer contains appropriate units for business processes.
4. **Infrastructure Services**: This layer includes functionalities such as service monitoring, service management, authentication, logging and security. Some of these functionalities are usually provided by an Enterprise Service Bus (ESB).
5. **Component-based service realizations**: This layer provides service implementations based on existing applications contained in the Operation Systems layer.
6. **Operational Systems**: This layer includes enterprise applications such as Customer Resource Management (CRM) systems, Enterprise Resource Planning (ERP) systems, databases, packed applications and legacy applications

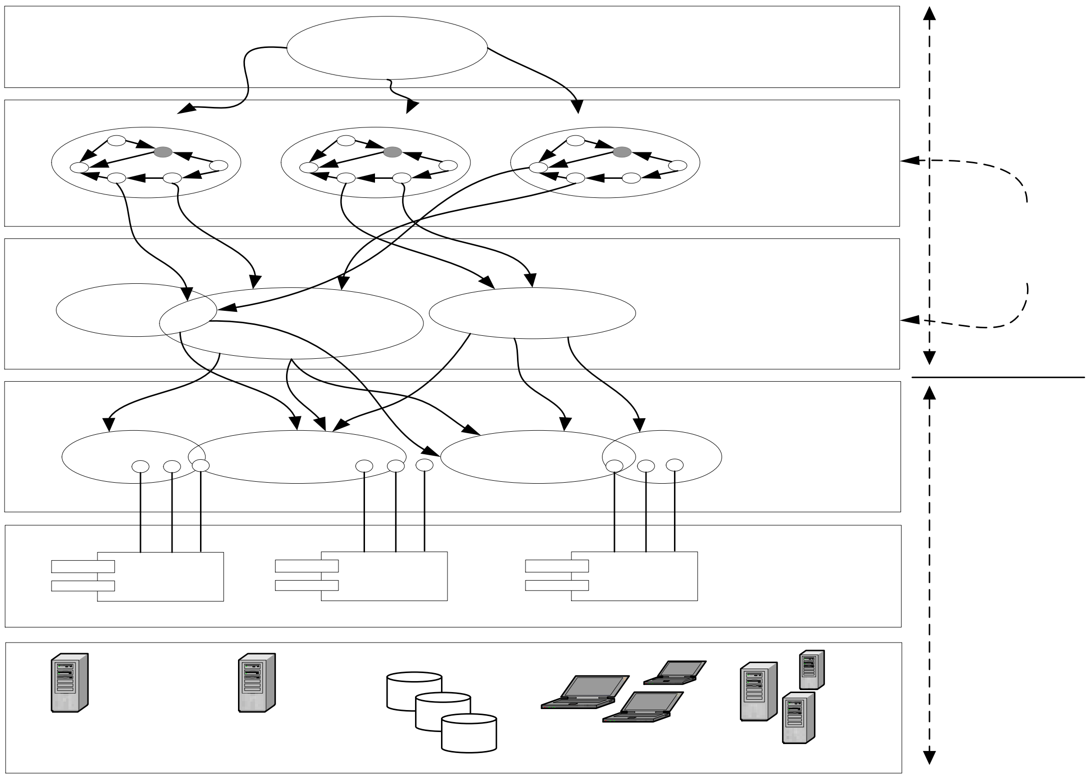


Figure 11: SOA layers for abstract functionality in an Enterprise Architecture [84].

A brief description SOA Concepts

Service is the main concept in SOA and identifying services in an organization is vital to ensure a successful enterprise solution. A Service can be described as a well-defined, encapsulated, reusable, business aligned capability [147]. The issues related to service design can be addressed using the "service-orientation principles "described by Thomas Erl [58]. These principles address issues such as "How services should be designed?", "How messages should be designed?" and "How should the relationships between services be defined?" [58]. A brief description of these principles is listed below:

- **Loose coupling**: Minimizes dependencies and relations between services.
- **Service contract**: Defines an agreement that describes the service and terms of information exchange.
- **Autonomy**: The service has control over its own logic implementation.

- **Abstraction**: The service implementation logic is hidden from the service consumers.
- **Reusability:** The services are designed in ways that support potential reuse.
- **Componsability:** Services can be used to build other services.
- **Statelessness:** Services are designed in a way that minimizes retention of information related to specific activities.
- **Discoverability:** Services can be discovered and assessed by service consumers.

Methodologies for implementing SOA based solutions

Several methods and techniques have been developed in order to effectively incorporate the "service-orientation principles" and address challenges such as composing and combining new services during the development of SOA-based projects [147]. Some of the most important methods for developing SOA-based solutions include the Service Oriented Modelling and Architecture (SOMA) from IBM [147], Service Oriented Analysis and Design (SOAD) also from IBM [6] and Thomas Erl's methodology [58]. Other significant efforts that address the complexities of SOA are the standards SOA-RM and the Service Component Architecture (SCA) [65] maintained by OASIS. The SOA-RM [144] is a reference model that provides definitions and vocabularies at a high level of abstraction that applies to all SOA and SCA is a set of specifications specifically designed to build distributed applications based on SOA. SCA represents the next step in the evolution of SOA, raising the level of abstraction and addressing two critical issues of software development namely complexity and reusability. SCA hides the complexity such as specifying security, reliability and other quality of service elements from the application code. SCA was originally developed by big vendors such as IBM, SAP, and Oracle among others, and it has gained increasing attention by developers of SOA solutions. SCA is reviewed in more detail in the next section.

3.7. Service Component Architecture

SCA is a set of specifications that provides a model to build SOA-based applications. The SCA effort started in 2005 by a group of vendors that includes IBM, Oracle, SAP, and others and handed over to OASIS in 2007. Some of the open source implementations include Apache Tuscany [67], Fabric3 [68], FraSCAti [69] and Red Hat SwitchYard [70]. In SCA, services are wired together in order to build an application. Services can be implemented using different technologies and programming languages such as Java, C++ or specialised languages like the Business Process Execution Language (BPEL). The main specification is the SCA Assembly model, which provides a standardized XML representation to define the configuration of the SCA application. SCA defines a model for the SCA component (figure 12) and a model for the SCA composite (figure 13).

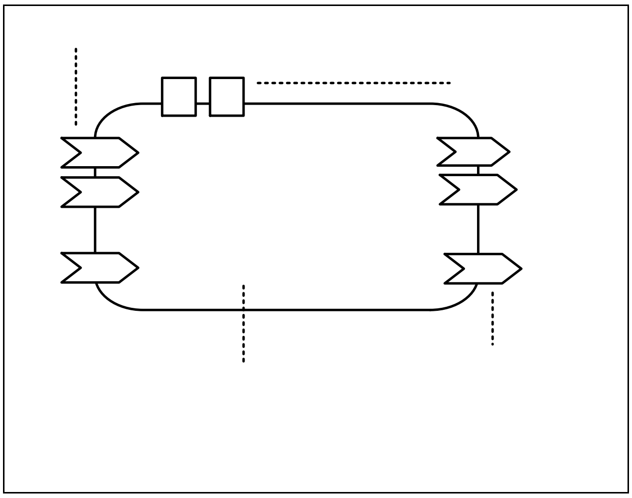


Figure 12: SCA Component Diagram [65]

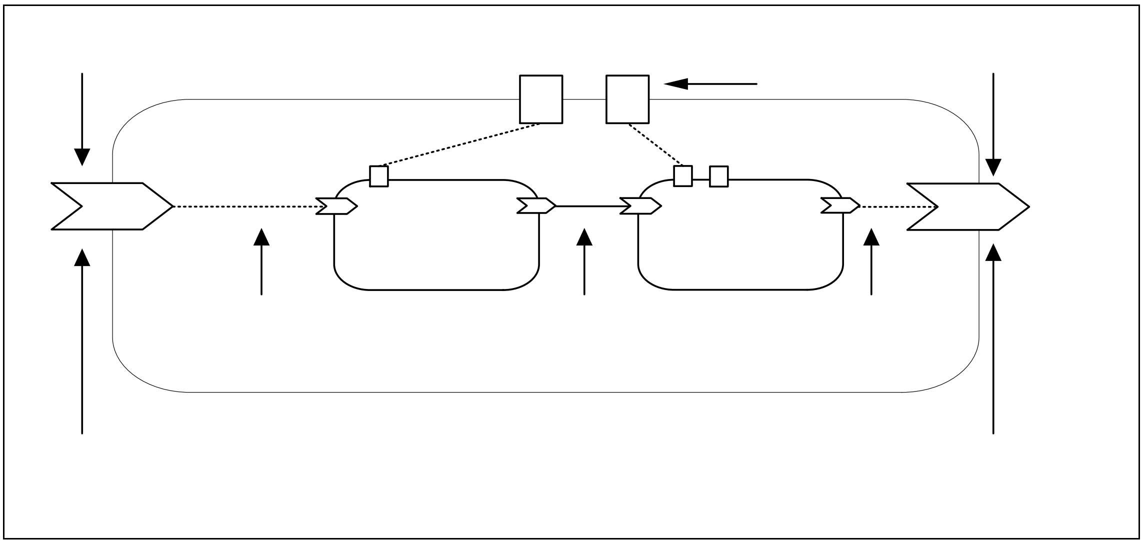


Figure 13: SCA Composite Diagram [65].

As is shown in figure 12, SCA components provide functionality through "services" and their dependencies are defined through "references." Properties allow configuration of the application using external values. Different programming languages can be used to implement the logic of the SCA components. Figure 13 shows how component A is connected or "wired" to Component B in order to create a composite service. Inside the SCA composite, services can be promoted, which means that the functionality of the SCA composite service will be provided by the promoted service. Similarly, references can be promoted, which define dependencies with external resources. The composite services and references define how the SCA composite will be exposed and connected to dependencies. Finally, composite applications are deployed in a SCA runtime, which is usually a part of an ESB [10]. The ESB concept is explored in more depth in the next section.

3.8. Enterprise Service Bus

The ESB is a software pattern that can provide the necessary infrastructure to support service orientation[7], [9]. ESB acts as an intermediary, routing, transforming or enriching messages between service providers and service consumers. As mentioned in section 2.4.3, the ESB is often used to provide the "infrastructure services". This is because the ESB can support various patterns as is shown in figure 14 [148].

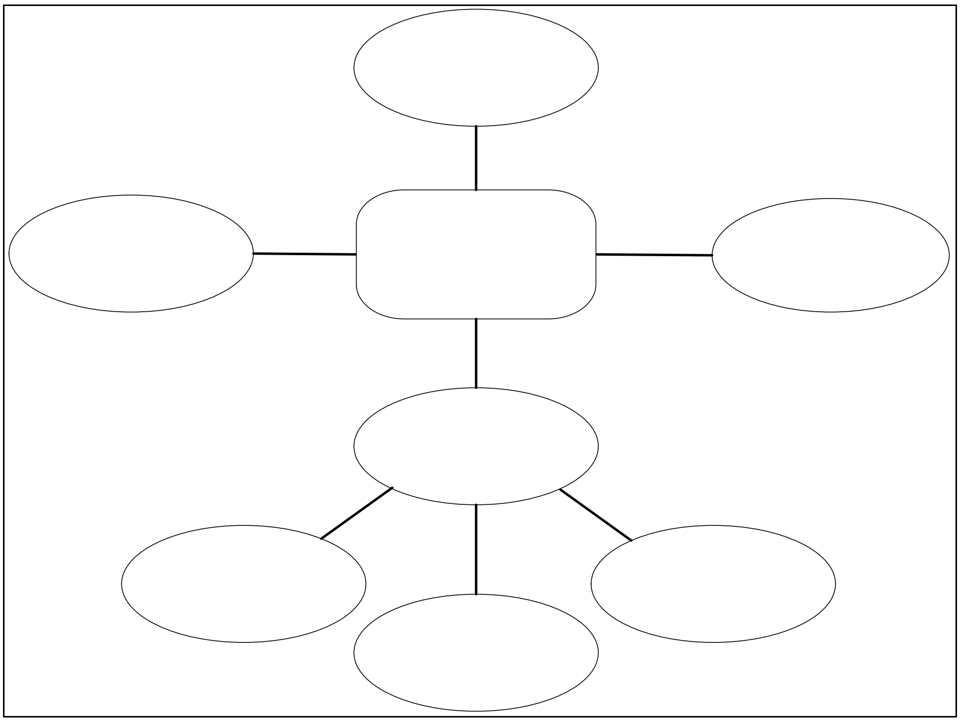


Figure 14: Some of the patterns and sub patterns supported by the ESB (adapted from [148])

Some of the patterns and sub-patterns supported by the ESB are briefly described below:

- ***Intermediate routing***: Based on routing logic allows definition of the source and destination to forward request and response messages.

- ***Event-driven messaging***: Most of the ESBs provide publish/subscribe mechanisms that allow subscribers to receive notifications of events that occur at the boundaries of the ESB.
- ***Asynchronous queuing***: Allows the exchange of asynchronous messages, which increase the reliability of message delivery.
- ***Service broker***: Includes patterns such as communication protocol bridging, data model transformation and data format transformation, which enable integration with disparate systems.

3.9. Business Process Management

In an SOA based enterprise architecture, complex services resulting from various business interactions take place in the functional SOA layer defined. These interactions, "business processes" are described in detail in section 2.5. Business Process Management (BPM) is a well-established discipline that "combines knowledge from information technology and knowledge from management sciences and applies this to operational business process" [149]. In SOA, business processes are usually accomplished through Web service composition using BPM languages such as the Business Process Execution Language (BPEL) or Business Process Model and Notation (BPMN). The combination of SOA and BPM technologies enables the organization to build executable business processes that span multiple systems and organizations. It also allows continuous monitoring and optimization processes to be supported. All of these benefits improve business agility and result in significant costs savings [150]. In the context of heath care, clinical practice that describes the sequential flow of clinical activities can be represented using business processes. Clinical guidelines describe flexible collaborative processes, structuring them into activities (tasks), which at times require human intervention [120]. Business process technology can be applied to consider all the guideline aspects: tasks, execution flows, participants, conditions, scheduling and temporal properties. Moreover, these models could also consider other temporal and management issues related to actor unavailability (e.g., vacations, holidays, and periodic work shifts and of physicians and nurses). For this reason, business process technology can be considered a suitable approach to manage all the guideline aspects, further enriching process and related information [120]. Recently, the BPMN language has become the preferred standard for business process modelling in industries like manufacturing and financial processes, [72].

Furthermore, BPMN 2.0, which is the latest version released in January 2011, allows modelling of different sets of business processes such as orchestration and collaboration and also extends the definition of human interactions [71], which are important concepts in healthcare. The BPMN standard is described in more detail in section 2.8.1.

3.9.1. Business Process Model and Notation (BPMN)

The primary goals of BPM models are three-fold: 1) document how the organization operates, 2) analyse processes in order to improve them, and 3) coordinate the execution of the process through the deployment of the model into a business process management system [151]. Business Process Model and Notation (BPMN) is a standard maintained by the Object Management Group (OMG) that can achieve the above mentioned goals. It provides a graphical notation that can be understood by domain experts and technical developers and has been adopted by the majority of software workflow suites [72]. Although BPMN has a large set of symbols, it has been observed that most business process can be described using a small portion of the BPMN constructs [152]. Figure 15 is a BPMN model that describes a hypothetical process for an insurance claim.

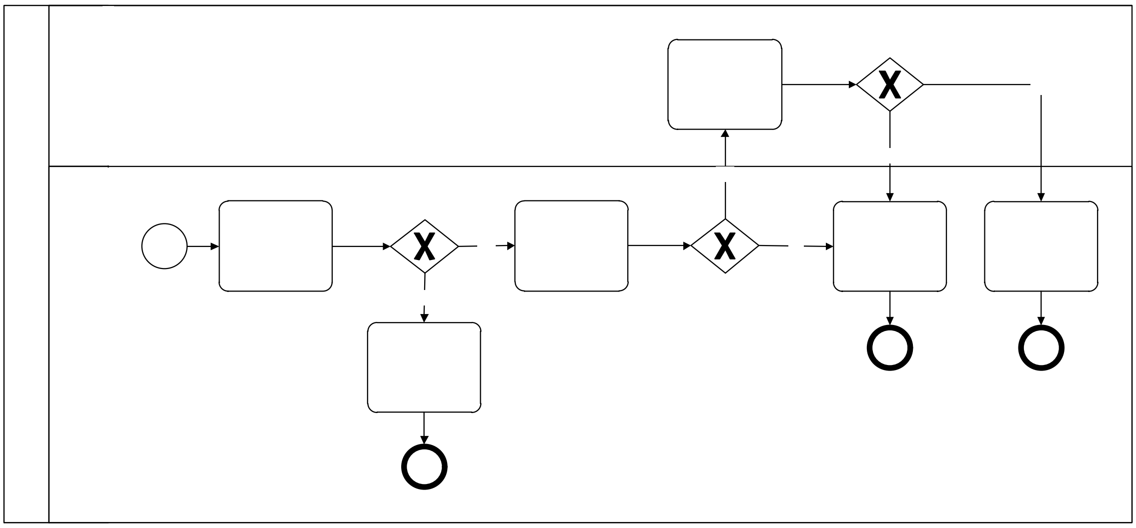


Figure 15: Hypothetical process for insurance claims

The BPMN construct *Pool* is used to describe the insurance claim process that takes place inside of a Medical Insurance Company. The actors involved in the process are represented by *Lanes* (i.e., Medical Insurance Claim System and the Insurance Manager). The process is initiated by the “Receive Claim” that is represented by a *start event*. The

Activities are symbolized by round boxes. The activity "Validate Data" is responsible for verifying data such as patient SSN, Health provider identification and policy number. If the system detects some inconsistencies in the data, it will generate a validation issue report and the process is terminated. If the data provided is correct, the system will determine the insurance coverage. If the insurance coverage is less than $10,000, then the system will automatically adjudicate the claim, otherwise the insurance manager will have to review the insurance claim and provide an approval. If the approval from the manager is received then the system will adjudicate the claim, otherwise it will generate an approval issue report and the process is terminated. As is shown in figure 15 the diamond shaped boxes with an X inside, are used to define that the process will take only one route (i.e., Yes or No). The end of the process is represented by the *end event* (i.e., circle with a thick line).

The various notations of a BPMN Process model are described in detail in Appendix B.

Usually, important business logic, which is encapsulated inside of the *Activities* of the business process model, is defined using Business Rules (BR) technology. Section 2.8.2, describes the concept of BR in more detail.

3.9.2. Business Rules

Automating business process through IT provides advantages such as increased productivity and rapid access to information. However, during the development of the information systems, important regulations, policies and core decision logic of the business are often embedded into the code of various systems distributed across the origination [153]. This allows the organization to identify and verify if process relevant rules have been implemented and whether they have been followed consistently [153]. This problem also allows domain experts to identify new business opportunities and make quick changes in order to meet the requirements of the organization [153]. In order to avoid these issues, regulations, policies and business logic can be extracted from information systems using Business Rules (BRs) technology. A Business Rule (BR) can be described as a "*statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business. The business rules that concern the project are atomic – that is, they cannot be broken down further*" [154]. BRs are expressed in a language that all stakeholders can understand and are usually centralized and managed in a Business Rules Management System (BRMS) [155]. Some of the most common formats for authoring business rules include: if-then rules, decision tables, decision trees, scorecards and custom languages [156]. The If-then format is the preferred type of language for condition-action rules [156].

For example, the *Validate Data* activity of the processes showed in figure 15 of section 3.9.1 can be expressed using if-then type rules. Assuming this activity includes the verification that the service provider is in the list of service providers for the insurer. This business rule can be expressed as follows:

In the healthcare insurance scenario described above, the example for function Validate Data describes the if-then logic is described below:

Logic Description

The claim form received by the application consists of data like: Patient's Social Security Number (SSN), Health provider identification and insurance policy number. For the activity Validate Data to execute, it must check if this information exists in a valid form and then proceed to the next step. The service Validate Date checks this information in the following manner

Logic expressed as Rule

If

Patient's Social security Number *IS VALID*

Health provide identification *IS VALID*

Insurance policy number *IS VALID*

Then

The Claim IS Valid

Once it has been established that the claim is Valid, the process moves to the next step.

3.9.3. Combining the SOA and BPM paradigms

Service oriented architecture is founded on the principles of reusable federated web services. The advantage of embedding business functionality in a web service is that it builds up a repertoire of services, which can be exposed to external applications and hence allow reusability of core business functionality.

From an architectural layer point of view, web services enabled by SOA are available to be accessed as either composite applications or standardised web interfaces by external client applications. The capabilities of SOA web services down to the functionality level can be accessed by applications residing at the integration level of the architecture. These services can then be invoked as business processes to enable a BPM and SOA based enterprise architecture. Secondly, the tasks embedding the business logic in a BPMN model can be executed using a business rules engine

3.10. Human Interaction workflows

In order to execute organizational processes efficiently, it is important to enable an organizational setting where all healthcare staff and other stakeholders are aligned around the same objectives (e.g., provide health care, reduce accidents, etc.) and work in a collaborative manner. Creating such an environment requires allocation of responsibilities among healthcare professionals and support staff, and also different divisions working towards the same goal. Supporting organizational and clinical processes is one of the main roles of modern IT architectures. In fact, continuous process improvement resulting in valuable outcomes would not be possible without help from IT [157]. Workflow technology has proven to be effective in supporting processes where humans are actively involved and interact with information systems [158]. The workflow technology that supports these types of processes is also known as “human interaction workflows” [158]. Three core components of human interaction workflows are:

Tasks: Human interaction workflows usually perform a set of activities that are part of a larger process that is composed of automated and non-automated tasks.

Role: Each activity in the human interaction workflow is assigned to a role, which represents a group of employees with the ability to carry out that specific task. The definition of roles provides some flexibility to the organization, since several employees are capable of performing the task.

Task List: An important concept in human interaction workflows is the list of tasks represented in a graphical user interface. This allows employees to directly interact with the process. For example, each employee can login into the system and review the list of tasks assigned to their role and claim the task that he or she wants to perform. The data that is produced as a result of this task is then evaluated by the system, which determines the next step in the workflow.

Several organizations have recognized the importance of human interaction for supporting the business process in SOA-based systems and have created standards that define the interaction between the human and the system. Some of these standards are described in section 3.11.

3.11. SOA Standards for human interaction

SOA has proven to be an ideal way of developing software to support processes. However, it is important to take into account that in a complex setting such as healthcare, services are provided not only by systems, but also by humans. Implementing these human task enabled services depends upon a number of factors, the most important one being providing results according to the availability of resources. Secondly, as humans work at varying speeds, the time taken to provide the resultant data from these human task enabled services may vary considerably. The SOA platform must support asynchronous communication and service composition using services provided by systems as well as human services [159]. The need for human interactions has led to the development of a number of specifications described below:

1. **BPEL4People:** BPEL4People is an extension of the specification BPEL 2.0 [160] that facilitates orchestrated interactions between Web services but does not include the concept of human interactions[161]. Some of the concepts introduced by BPEL4People include:
    - Generic process-related human roles (process initiator, process-related stakeholders and business administrators).
    - Mechanisms to assign people to activities.

- "People activity" which allows definition of human interactions within BPEL models.
- Distinguish between local and remote human tasks.

2. **WS-HumanTask:** BPEL4People is closely related to WS-HumanTask, which was also developed by the same technical committee [162]. The specification WS-HumanTask defines the notion of human tasks, which is used to specify work that has to be carried out by people and allow the integration of humans in SOA-based applications. The following are some of the features defined by WS-HumanTask:
    - Assign people to tasks (i.e., roles such as task initiator and task stakeholder)
    - Ad-hoc attachments or comments to tasks.
    - Defines two interfaces for human tasks: 1) Expose service provided by the human task, and 2) Allow people to deal with tasks.
    - Setting timeouts for tasks and actions to be taken if the timeout expires (i.e., escalation actions).
    - Specify how the human tasks may be rendered on different applications or devices.
    - Define notifications (i.e., missed deadlines).

Together, these specifications rely on other standards such as XSLT, Infoset, XPath, WSDL, XML Schema, WS-Addressing, WS-Coordination and WS-Policy. WS-HumanTask is not tied to BPEL. Several vendors have implemented this specification in business process engines based on the business process language BPMN, which also supports human interaction and is the preferred standard for business process modelling [72]. BPEL is primarily oriented to technical users in contrast to BPMN, which provides a graphical notation that can be understood by domain experts and technical developers [72]. The simplicity of the BPMN graphical notation makes it a good option for modelling processes in the healthcare domain.

3.12. Summary and conclusions

From the results of the literature review described in chapter 2 and in this chapter, the following identified issues will serve as a basis for this study:

Requirements for SOA based Architecture for CDS

- There is a need for an architecture model for sharing executable CDS modules and services.
- Provide an interface that facilitates domain experts to understand the decision making logic and also provide the mechanism to make quick changes in a process.
- Clinical workflows designed to support an HCO must address the diverse needs of factors such as the complexity of clinical scenarios, clinical settings (scale of the HCO) and the institution's strategic goals. The literature survey and past experiences with HIT implementations indicate that it cannot be a one-size-fits-all solution; rather a decentralized approach must be adopted for this implementation to address the multiple requirements of an organization.
- The architecture has to be designed keeping in view a plethora of data models used by multiple information systems in an organization. In order to make the exchange of data a robust process, the data models must be viewed in detail at the design level in order to provide long-term benefits.
- It has been inferred from the literature review that interoperability is one of the key challenges when building a HIT infrastructure. This not only includes the IT solutions for clinical decision support but other systems such as legacy systems, PAS, LIS etc. Implementing systems that have interoperability as a core feature must be a part of the strategic goal of implementing an HIT in an organization. Having an effective communication mechanism across applications will bring together independent systems and legacy systems.
- In a large scale enterprise healthcare architecture, various vendors are involved in providing specific solutions. These solutions may be deployed locally or in the cloud. This gives rise to a distributed infrastructure for HIT. The organization must allocate sufficient resources to manage the distributed nature of IT systems that together deliver the goals of the HCO.

Propositions for an SOA based architectural solution for CDS

- Approach for development of complex HIT systems
  - Adoption of an EA framework like TOGAF.
  - Adopt EA modelling languages such as ArchiMate

- Development of a standard-based service oriented architecture that will allow access to healthcare data from multiple distributed healthcare systems.
    - SCA: Using a SOA-based programming model will enable enterprise functionalities to be exposed as web services, thus allowing reusability by other applications desiring the same functionality.
- BPMN
    - Simplistic view of complex interactions among various actors in a clinical and non-clinical workflow.
    - Support of human interactions in addition to automated systems for the purpose of decision support.
- Business Rules
    - Business Rules expose business logic in a human readable format,
    - Feasible for embedding logic for clinical decision support in the form of rules.
    - Allow domain experts to make quick changes to address organizational agility.
- Integration of HIT Tools and Technologies using enterprise service bus
    - Allows interoperability between different systems.
    - Development of adapters for different healthcare standards.
    - Allows integration of different CDS services that may or may not be a part of the EHR.
- Standard interfacing with EHR to provide CDS results to the physician at the point of care.

The propositions mentioned above will be used as a basis for developing an integration solution for providing CDS at the point of care. The case studies testing the feasibility of this architecture solution will follow in Chapter 4 for WGS enablement for CDS. Chapter 5 tests this architecture for Diagnosis and Urgent Referral (COPD, and Lung Cancer). Finally Chapter 6 tests this architecture for Organizational practice.

Chapter 4

Whole Genome Sequence Guided Clinical Decision Support

4.1. Introduction

This chapter describes the design, development and evaluation of an architecture that supports WGS-guided CDS services. This chapter addresses the research objectives described in Chapter 1. The architecture includes a CDS Web service, EHR, WGS data service and an ESB. The function and interaction of each of these components is described through a clinical scenario of a hypothetical patient with high risk for Lynch syndrome based on genomic information. Since the first draft of the human genome was published in 2001, the area of DNA (Deoxyribonucleic Acid) sequencing has experienced tremendous progress and continually cheaper and more rapid methods are developed [163]. Whole Genome Sequence (WGS) is one of the technologies that offer identification of gene variants that cause disease. This type of information will be available shortly for use in clinical settings and will enable physicians and other health professionals to provide personalized medicine [163]. The concept of personalized medicine is an important growing field of healthcare that uses a person’s unique clinic, genomic and environmental information to optimize health care [164]. Several studies have shown that personalized medicine can improve health care considerably as each person reacts differently to medications and treatments and these differences are often anticipated by identifying genetic variations [165]. Despite the benefits of the use of genetic information in the clinic, there are several barriers that must be overcome before this becomes a reality. Some of these barriers include: 1) limited knowledge of genetics among physicians, 2) complexity of genome information, 3) and lack of experts in genetics in clinical environments [166]. CDS offers a potential solution to overcome these barriers facilitating the use of WGS information during routine patient care [166]. Nevertheless, due to continuous advances in genome interpretation and the complexity of the information, there are several aspects that must be considered when developing CDS based on WGS information [167]. Moreover, to be effective, CDS must be delivered within the clinical workflow at the point of care and at the time of decision making [166].

4.2. Whole Genome Sequence

WGS provides detailed information about the genomic variations for a specific person. These variations are differences in the sequence of DNA from one individual to another.

WGS information can be used to identify pathogenic variants in disease-causing genes and assist in the diagnosis process. The following table shows some of the potential examples identified by Welsh et al., where the use of WGS information can enable CDS [168].

Table 11: Potential examples of use of WGS information to enable CDS [168].

CDS functionality type	**Clinical genomics example**
Medication dosing support	CDS automatically adjusts warfarin dosing as a result of known alleles in the VKORC1 and CYP2C9 genes.
Order facilitators	An order for colonoscopy is recommended at a younger age as a result of known pathogenic mutations in genes associated with colon cancer.
Alerts and reminders	During medication ordering, gene variants known to affect drug pharmacokinetics are checked and clinicians are alerted to potential gene-drug interactions.
Relevant information display	Context aware infobuttons in the problem list leverage genome data to provide genetic risk information for a patient with breast cancer.
Expert systems	The EHR provides a 10-year cardiovascular disease risk score based on clinical, environmental, and genetic risk factors.
Workflow support	The EHR schedules a generic counselling consultation during prenatal visit due to the presence of an X=lined disease gene variant.

4.3. Technical aspects to consider when using genome data with CDS

Due to the complexity and volume of the genomic sequence information, there are several aspects to consider before integrating with clinical systems. Masys et al., defined a list of technical features that should be supported in order to efficiently integrate genomic information with EHRs [169]. Later, this list is extended by Welsh et al., in order to fulfil the specific requirements of CDS [168]. Table 12 shows the complete list of requirements for the integration of genomic data with CDS. Each of these technical features is briefly discussed below.

Table 12: Technical requirements defined by Masys et al. [169] and Welch et al. [168].

Desiderata for the integration of genomic data into EHRs defined by Masys et al. [169] 1. Maintain separation of primary molecular observations from clinical interpretations of those data 2. Support lossless data compression from primary molecular observations to clinically manageable subsets 3. Maintain linkage of molecular observations to the laboratory methods used to generate them 4. Support compact representation of clinically actionable subsets for optimal performance

5.	Simultaneously support human-viewable formats and machine-readable formats in order to facilitate implementation of decision support rules
6.	Anticipate fundamental changes in the understanding of human molecular variation
7.	Support both individual clinical care and discovery science
Additional desiderata for the integration of genomic data with CDS defined by Welch et al. [168].	
8.	CDS knowledge must have the potential to incorporate multiple genes and clinical information
9.	Keep CDS knowledge separate from variant classification
10.	CDS knowledge must have the capacity to support multiple EHR platforms with various data representations with minimal modification
11.	Support a large number of gene variants while simplifying the CDS knowledge to the extent possible
12.	Leverage current and developing CDS and genomics infrastructure and standards
13.	Support a CDS knowledge base deployed at and developed by multiple independent organizations
14.	Access and transmit only the genomic information necessary for CDS

Maintain separation of primary molecular observations from clinical interpretations of those data – Laboratories commonly report the genetic variations following the practices of pathology and radiology, where just a small number of observations are cited along with a professional interpretation. Since genomic science is continually evolving, there is a need to separate the primary observations from their interpretation in order to be able to update the interpretations at a later date.

Support lossless data compression from primary molecular observations to clinically manageable subsets – Given the importance and volume of the genetic data (range from gigabytes to terabytes in its raw form), it is necessary to use lossless data compression techniques.

Maintain linkage of molecular observations to the laboratory methods used to generate them – There are different technologies to generate the genetic sequencing. Each of these technologies has advantages and disadvantages, for example, some biological phenomena are not detected by certain methods. For this reason, it is important to keep information about the method used to obtain the molecular observations.

Support compact representation of clinically actionable subsets for optional performance – Use of codes or keywords that represent a physiological state, rather than continually query the entire genome. For example, the patient presents a minor allele variant CYP2C19*2, which is associated with altered metabolism of commonly prescribed drugs.

Simultaneously support human-viewable formats and machine-readable formats in order to facilitate implementation of decision support rules – The research literature about the relationship between the genetic variations and the clinical phenomena is continually increasing, this makes it difficult for physicians to stay updated. However, the use of decision support rules can help physicians to receive current patient-specific information via alerts or reminders.

Anticipate fundamental changes in the understanding of human molecular variation – The design of the EHR must take into account that the genomic information of an individual is not immutable. Studies show that the genetic information can change due to aging or mutations caused by diseases such as cancer.

Support both individual clinical care and discovery science – Well-structured genetic information within the EHR would allow this information to be used to support research in genomic discovery science.

CDS knowledge must have the potential to incorporate multiple genes and clinical information – In order to provide an accurate assessment, the CDS knowledge base must be able to incorporate different pieces of genetic information as well as clinical information (e.g. age, weight, health history, co-morbidities) and environmental factors (e.g. diet, physical activity, stress).

Keep CDS knowledge separate from variant classification – The interpretations of the genetic variations are frequently changing due to advances in genetic science. Thus, separating variant interpretations from the CDS knowledge base would facilitate the management of the continuous changes in the variant interpretations without having to make changes in the CDS knowledge.

CDS knowledge must have the capacity to support multiple EHR platforms with various data representations with minimal modification – In order to be able to share WGS enabled CDS across multiple different healthcare organizations, the architecture of the CDS should not be tied to a specific clinical system.

Support a large number of gene variants while simplifying the CDS knowledge to the extent possible – Since there is a large amount of variants per gene and continually new variants are discovered, it would be inefficient to try to maintain CDS knowledge for every variant. For this reason, it is advisable to classify variants with similar clinical impact.

Leverage current and developing CDS and genomics infrastructure and standards – There are a number of standards and infrastructures that support the use of genetic information and CDS, which can be used as a basis for the development of new standards.

Support a CDS knowledge base deployed at and developed by multiple independent organizations – Since the time and cost of development of CDS knowledge can be prohibitive for a single organization, there is a requirement to adopt an approach that allows private or public organizations and individuals to easily modify and distribute CDS knowledge.

Access and transmit only the genomic information necessary for CDS – Transmitting the entire genome would be inefficient and insecure. Thus, it is recommended to only provide relevant information for CDS.

4.4. Clinical Scenario: Lynch Syndrome

4.4.1. Overview of Lynch syndrome

Lynch syndrome, also known as hereditary non-polyposis colorectal cancer, is a genetic condition that increases the risk of colorectal cancer. Lynch syndrome also increases the risk of other types of cancer such as stomach, intestines, liver, brain and skin cancers.

4.4.2. Genetic presentation of Lynch syndrome

Upon DNA analysis, patients who have pathogenic variants in genes such as MSH2, MLH1, MSH6 and PMS2 are at high risk of developing this condition [170].

4.4.3. Scenario for Implementation

In this section, a clinical scenario of a hypothetical patient with high risk for Lynch syndrome based on her genomic information is described. It is recommended that patients at high risk of Lynch syndrome receive screening colonoscopy every year or two years beginning at age 20 [171]. The following sections use this clinical scenario to demonstrate the functionality supported by the proposed architecture.

4.4.4. Requirements for Implementation

1. An EHR capable of recording Patient Demographic data (such as age)
2. A Web service to collect patient data
3. Rules authoring for Lynch Syndrome Testing
4. Genetic database of patients
5. Recording Genetic testing results for patients
6. Collecting genetic data for clinical decision support (recommendations for colonoscopy)
7. A Service Enabling CDS
8. Providing CDS recommendations at the point of care
9. Integration of CDS Rules with the EHR
10. Standards based implementation

4.5. Architecture Components

In order to implement the scenario explained in section 4.4.3, the developed CDS architecture consists of the following four components:

a) WGS data service

b) EHR

c) CDSS

c) ESB

Figure 16 shows an overview of the message sequence between the four components. The functionality of each of these components is described on sections 4.4.1 (WGS Service), 4.4.2 (ESB), and 4.4.3 (CDS Service).

These four components are graphically represented using ArchiMate modelling language. Some of the constructs of ArchiMate modelling language that are used for each of the components are described on Appendix C:

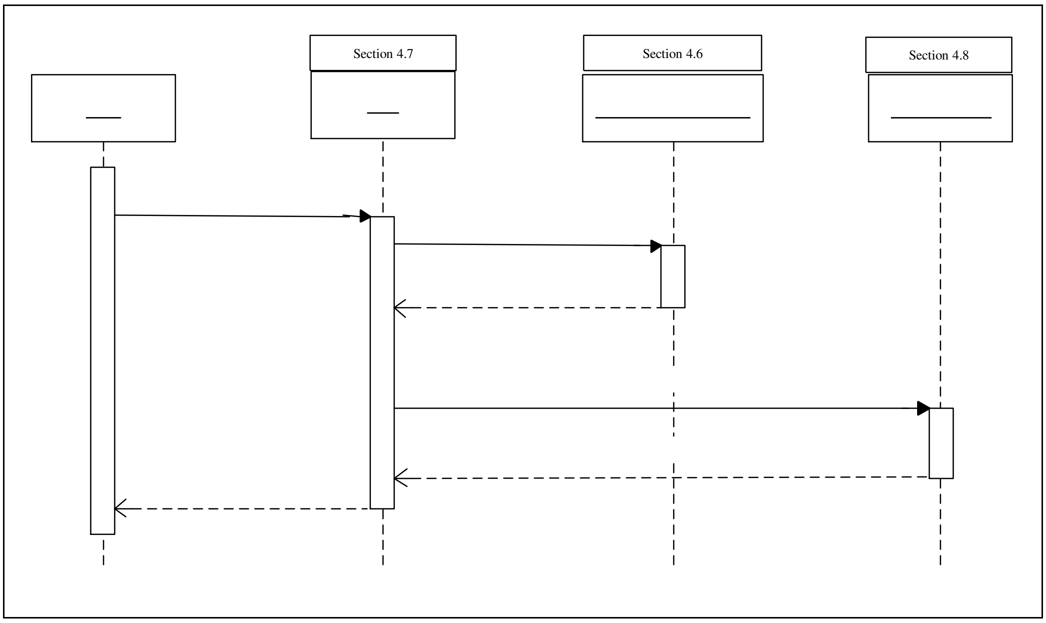


Figure 16: Sequence of messages between software components.

The EHR is the system that initiates the interaction by sending a request for decision support "getCDS()". The request is a Web service call based on the HL7 Clinical Decision Support Service (DSS) standard and consists of patient's information following the HL7 Virtual Medical Record (vMR) standard. After receiving the message request, the ESB verifies if the vMR contains the information as required by the CDS service. If the information is incomplete (e.g., lack of genome information), the ESB sends a data request to the appropriate system (e.g., request data from the Whole Genome Data Service). The missing information is then incorporated into the original vMR in order to submit a request to the CDS Service. The response of the CDS service is forwarded by the ESB to the EHR, which displays the recommendation to the physician as an alert.

4.6. WGS Data Service

To make patient's genetic information available to other applications, a Web Service was developed. This service is called "WGS data service" and consists of the following two components:

4.6.1. Genome Database

The Genome database is a relational database (MySQL). The database includes a table called 'patient_genome', presented in Table 13, containing the following columns (column names in quotes):

Table 13: 'patient_genome' table description.

Column name	Description
MRN	Medical record number which matches the patient id of the EHR.
gene	The gene where the variant was detected, this is expressed using HUGO (Human Genome Organization) Gene Nomenclature Committee (HGNC) standardized nomenclature.
refSNP	Reference SNP ID number assigned by the public archive for generic variation Single Nucleotide Polymorphism Database (dbSNP) [172].
nuc_var	Nucleotide variant according to the Human Genome Variation Society (HGVS) nomenclature [173].
pro_var	Protein variant defined using HGNC nomenclature.
Interpretation	Clinical impact of the genetic variation defined by the repository of human genome sequence variations ClinVar, which is supported by the National Center for Biotechnology Information of the U.S. National Library of Medicine [174].
id	Table primary key.

4.6.2. RLUS Interface

The information presented in table 13 (patient_genome) is then exposed as a SOAP Web service based on the Retrieve, Locate, and Update Service (RLUS) standard. RLUS is a Web service specification that was specifically designed to support SOA and allows retrieving, locating, and updating patient data resources across and within healthcare organizations [131].

The RLUS specification introduces the concept of "Semantic Signifier" in order to separate functional capacities of the service (retrieve, locate and update) from the semantic content. This allows the RLUS specification to support different information objects and metadata. For instance, RLUS can be used to transfer clinical documents that are based on standards such as Continuity of Care Document (CCD), Clinical Document Architecture (CDA) or Virtual Medical Record (vMR). For this implementation the standard vMR was selected since this standard was specifically designed to support CDS.

The Web service specified by the RLUS standard includes seven methods (*Describe, Discard, Get, Initiate, List, Locate* and *Put*). Because the main purpose of the application is to provide information about the genetic variations of a specific patient that is stored in a database, so for this implementation the 'Get' method is used. Figure 17 describes the interaction between external Web service clients and the RLUS Web service interface of the WGS Service data application.

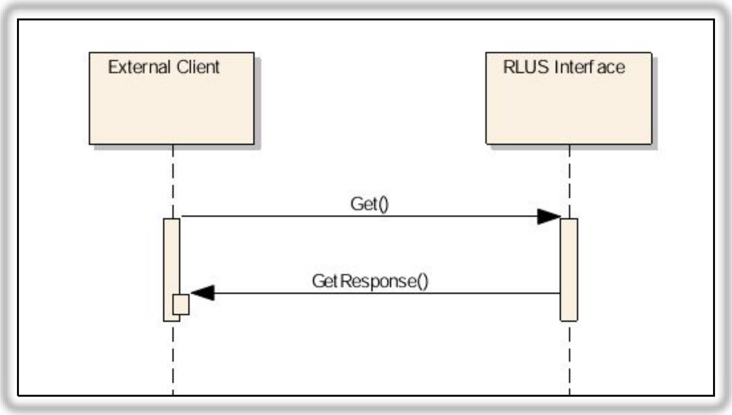


Figure 17: Interaction between the RLUS Web service interface and external client.

<u>Get() Request</u>

The Get() message request is defined through "RLUSSearchStruct", which is a structure that includes information such as sematic signifier and search criteria (e.g., Expression or Query-by-example). Figure 18 shows an example of a Get() message using a search criteria by "Expression".

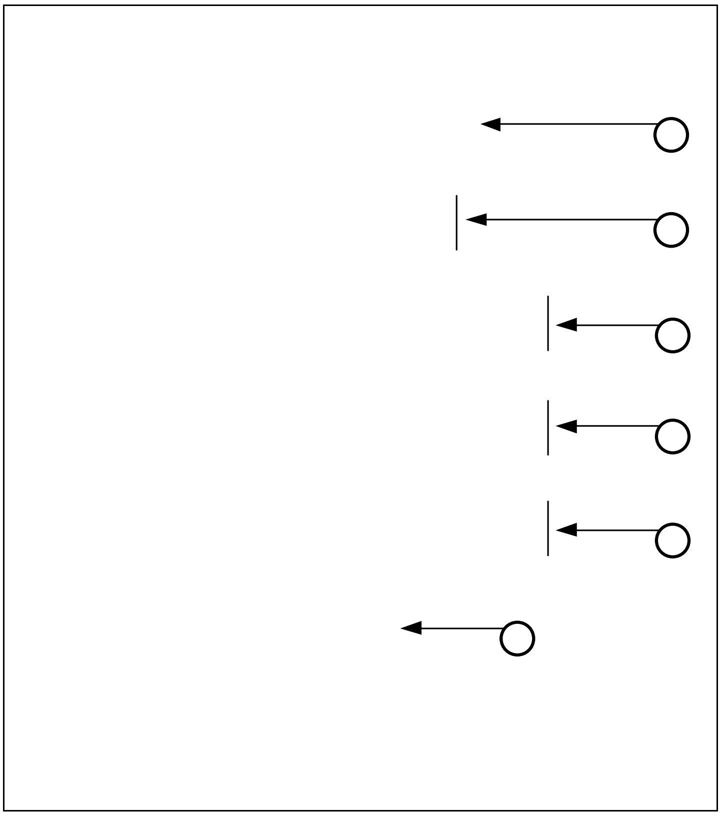


Figure 18: RLUS Get SOAP message request example.

As shown in figure 18, the following sections are described in detail below:

Semantic Signifier:

Defines the semantic-signifier, which must be a valid semantic name that indicates the underlying XML schema used to format the response that corresponds with the retrieved data from the retrieval process (1).

Filter Criteria:

The FilterCriteria includes four expressions that define a query to retrieve the genetic information of the genes ASC (3), MUTYH (4) and MLH1 (5) located in the vMR observationFocus element of the patient with identification number 33333 (2). The filter criterion is based on the "Equal To" operation.

Results Ordering:

Requests to present the results in ascending order (ASC) (6).

Get() Response

Figure 18 shows The Get Response() message of the request. It contains a vMR within the SOAP message body.

Figure 19: vMR representation of a genetic variation in a GetResponse Message.

Figure 19 shows a snippet of the vMR of the Get Reponse() message containing information of the variant of the gene MLH1. The vMR includes the observation value "NM_000249.3:c.982C" according to the HGVS nomenclature. The interpretation of "pathogenic" is expressed using the LOINC (Logical Observation Identifiers Names and Codes) system.

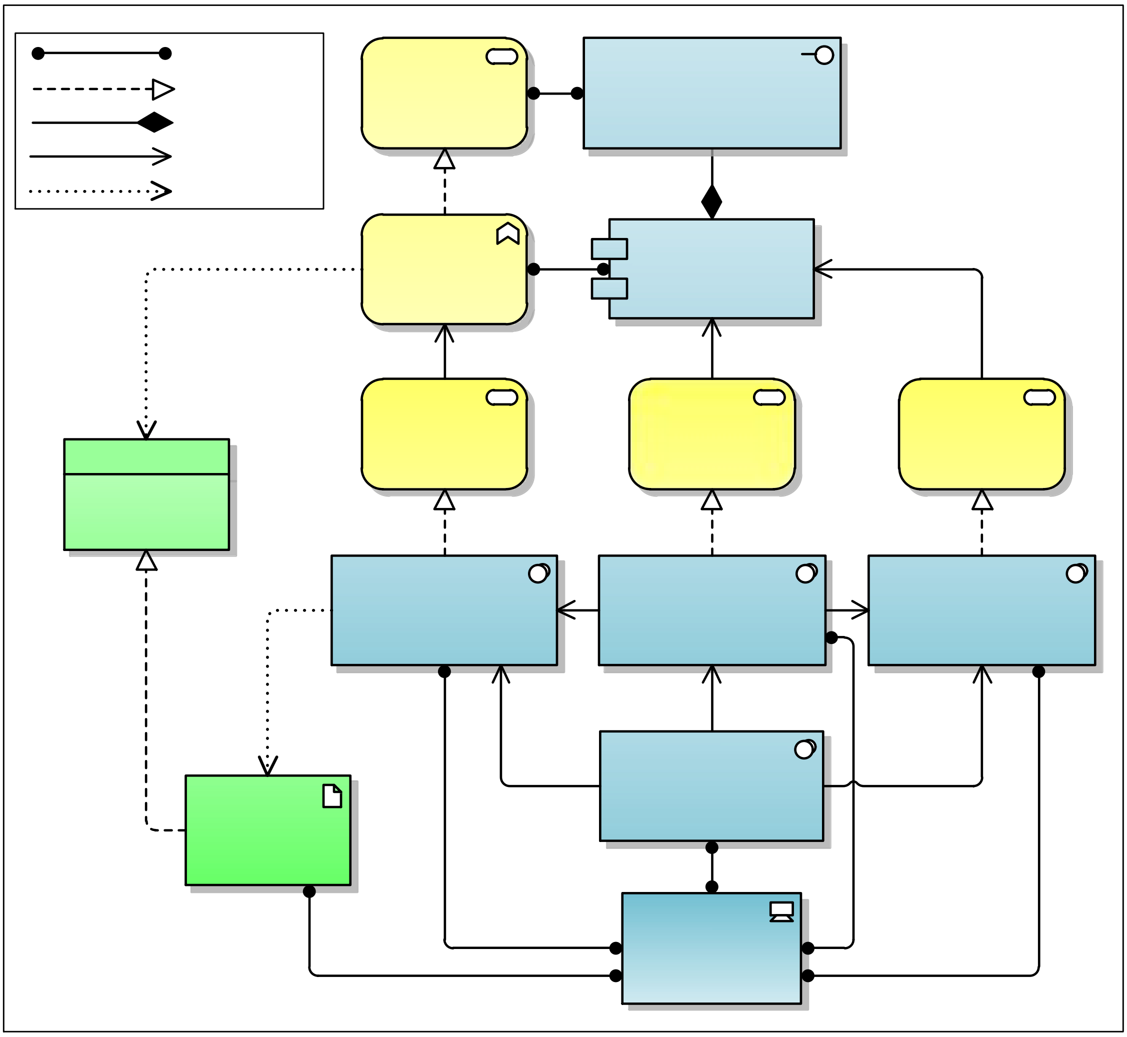


Figure 20: WGS data service application architecture.

Figure 20 shows an overview of architecture of the WGS data service application using the ArchiMate modelling language. The components used for this architecture are modelled and the relationships between the various components are presented The content of the labels follow the style defined by Gerben Wierda [175]:

- *First line*: Grouping information in brackets, for the infrastructure layer, usually the name of the device, e.g., [ubuntusrv1] and for the application layer the name of the application, e.g. [WGSDataService] and [WGSbd].-
- *Second line*: Any name assigned by the architect.
- *Third line*: Type of the ArchiMate element, e.g. (Device), (Infrastructure Service), etc.

The components have been grouped in [ubuntusrv1], [WGSdb] and [WGSDataService].

a. The [ubuntusrv1] includes a 64-bit computer running a Linux Ubuntu 12.04 LTS with the Relational Database Management System (RDBMS) MySQL Community sever 5.6.15, a JBoss Application Server version 7 and the Oracle Java Virtual Machine (JVM) version 1.7.
b. The [WGSdb] includes a database expressed as an artifact “WGS Data Base”, which realizes the data object “patient_genome”. The Infrastructure Services “JVM” and “Java EE Server” are used by the Application Component “WGS Application. The Infrastructure Service “MySQL Server” is used by the Application Function “WGS Data Access” that accesses the data table “patient_genome” and realizes the Application Service “RLUS Service”, which at the same time is assigned to the Application Interface “RLUS SOAP Web Service”.

Table 14 shows the hardware where the WGS Service was deployed and table 15 indicates the software used for this implementation.

Table 14: Hardware used to deploy WGS service.

Processor	Intel® Core™ i7 CPU 920 @ 2.67GHz × 8, 64-bit
Hard drive	950 GB
Memory	16 GB

Table 15: Software used to deploy WGS service.

Operational System	Linux Ubuntu 12.04 LTS, 64-bit
Database	MySQL Community Server 5.6.15
JVM	Oracle JVM 1.7
Application Server	JBoss Application Server 7

4.7. Enterprise Service Bus

The ESB is responsible for determining whether the message requests contain the information required by the CDS service and acts accordingly by collecting data from the corresponding system. The ESB is responsible for the Service Orchestration. Service

Orchestration can create new services by combining existing services within a business process, which in controlled by one of the participating systems [176]. Writing the logic to perform these tasks would typically require the intervention of a domain expert, because it is the expert who has the knowledge of what patient information is required and its location. Therefore, it was decided that the ESB should allow definition of the orchestration logic by using a graphical business process language such as BPMN, which can be understood by both developers and domain experts. Another, requirement was that the ESB should support the SCA standard in order to ensure that SOA best practices are followed. The figure 21 describes the configuration and main components of the ESB using the ArchiMate modelling language.

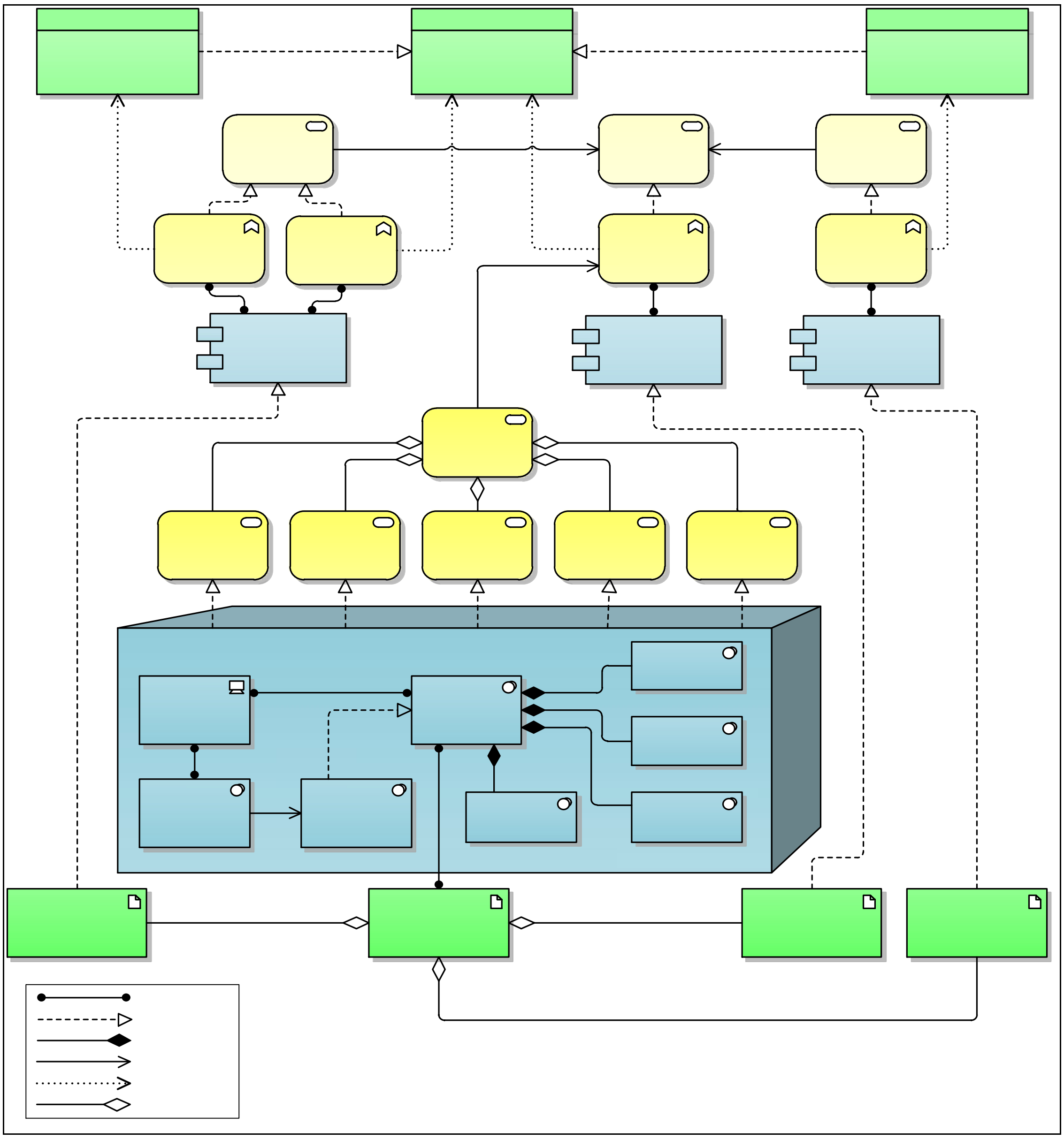


Figure 21: Archimate model for ESB Overview.

As shown in figure 21, the open source JBoss ESB SwitchYard was selected for the implementation of this prototype. As shown in the Archimate model of the ESB, The infrastructure services offered by this ESB include:

a. SCA runtime
b. Transport mechanism such as HTTP,
c. Business Process Engine,
d. Rule Engine
e. Apache Camel Integration Framework.

These services are provided by various open source technologies, which when working together, compose the SwitchYard ESB. The ESB was installed in an x64 machine running a Linux Ubuntu 12.04 LTS. To enable communication between CDS services and EHR systems, an adapter of the DSS standard was created called *ESB SwitchYard HL7 CDS Service Adapter*. This adapter is represented as an artefact, which realizes the Application Component "*HL7 CDS Service Adapter"*. This application component performs two functions.

a. CDS Service Provider
b. CDS Service Client.

The CDS Service Provider and CDS Service Client application functions realize the Application Service "*HL7 CDS Service*". The Application Function "*CDS Service Provider*" is responsible for receiving the Clinical Data (represented as Data Object) that come with CDS service requests. Similarly, for communication between the ESB and WGS data service the adapter "*ESB SwitchYard HL7 RLUS*", represented as an artifact, was implemented. This adapter realizes the Application Component "*HL7 RLUS Adapter*". The Application Function "*HL7 RLUS Get*" is assigned to "*HL7 RLUS Adapter*" that realizes the Application Service "*HL7 RLUS*". The Application Function "*HL7 RLUS Get*" receives the Genome Data (represented as a Data Object) that results from the WGS data service request.

The routing functionalities provided by the ESB are represented as the artifact "*ESB SwitchYard Routing Distribution*". This artifact realizes the Application Component "*CDS Routing*" to which has the Application Function "*CDS Routing Logic*" is assigned to. The Application Service "*CDS Routing*" that is realized by the Application Function

"*CDS Routing Logic*" uses the "*HL7 CDS*" Application Service to access the Clinical Data and verify if it contains enough information to use a CDS Application Service. If the information is incomplete, the CDS Routing Logic uses the "*HL7 RLUS Get*" Application Service to collect the missing information (Genome Data). Both data objects Clinical Data and Genome Data allow the CDS Routing Logic to create the CDS Input Object, which is then accessed by the Application Function "*CDS Service Client*" to send a request to an external CDS service. The hardware and software used to deploy the ESB is described in Table 16 and table 17 respectively.

Table 16: Hardware used to deploy the ESB.

Processor	Intel® Core™ i7 CPU 920 @ 2.67GHz × 8, 64-bit
Hard drive	950 GB
Memory	16 GB

Table 17: Software used to deploy the ESB.

Operational System	Linux Ubuntu 12.04 LTS, 64-bit
Database	MySQL Community Server 5.6.15
JVM	Oracle JVM 1.7
Application Server	JBoss Enterprise Application Platform 6.1
ESB	JBoss SwitchYard which includes: • Apache Camel • JBoss jBPM • JBoss Drools • SwitchYard SCA Runtime

The "CDS Routing Logic" provided by the ESB is described in detail in the next section.

4.7.1. CDS Routing Logic

The CDS routing logic is contained in the SCA composite shown by figure 22. This SCA composite includes the following four SCA components:

- *CamelServiceRoute*
- *ProcessComponent*
- *VeryGenomeDataServiceBean*
- *RequestGenomeDataServiceBean*

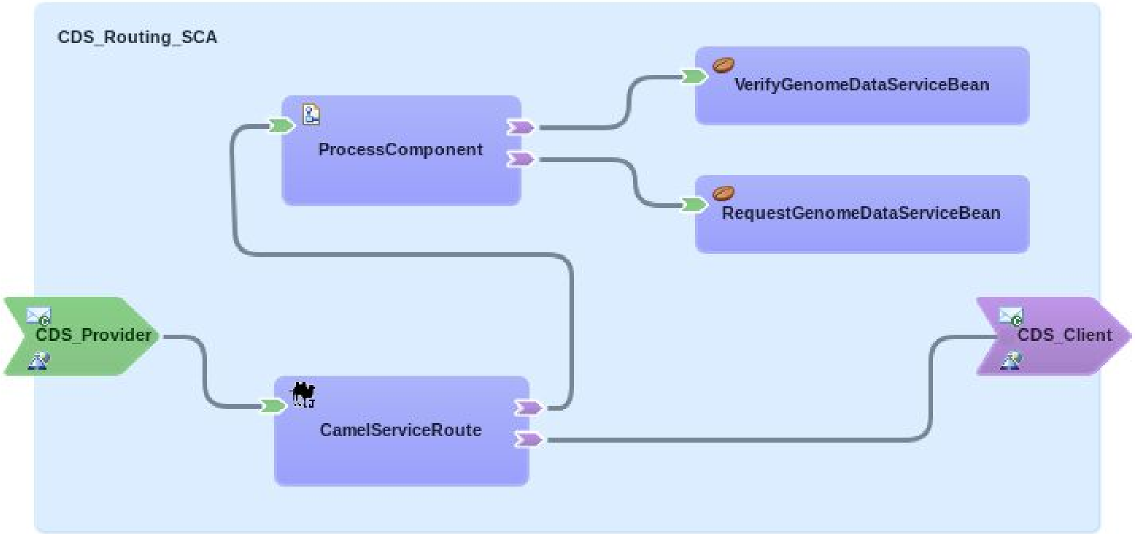


Figure 22: SCA Composite showing the CDS routing logic.

The *CamelServiceRoute* SCA component receives SOAP requests through the *CDS_Provider service* (green arrow, left), and transforms the contained vMR document into a Java object. This Java object is sent to the *ProcessComponent* SCA for evaluation. The Java object result from the evaluation is transformed into a vMR document and incorporated into a SOAP message request that is sent to the CDS Service through the interface *CDS_Client* (purple arrow, right). The *ProcessComponent* SCA contains a BPMN process (see figure 23) which is instantiated when new Java objects containing the vMR are received. The two SCA components wired to *ProcessComponent* perform tasks that are defined in the BPMN process. The first task in the process is to verify if the vMR contains genome data, if so, the process ends and the *ProcessComponent* SCA component returns the same vMR as the result. Otherwise, the task "*RequestGenomeDataServiceBean*" sends a request to the WGS data service and incorporates the missing information into the original vMR, which is returned as a result from the *ProcessComponent* SCA component. The task to verify the above condition is modelled using BPMN in figure 23.

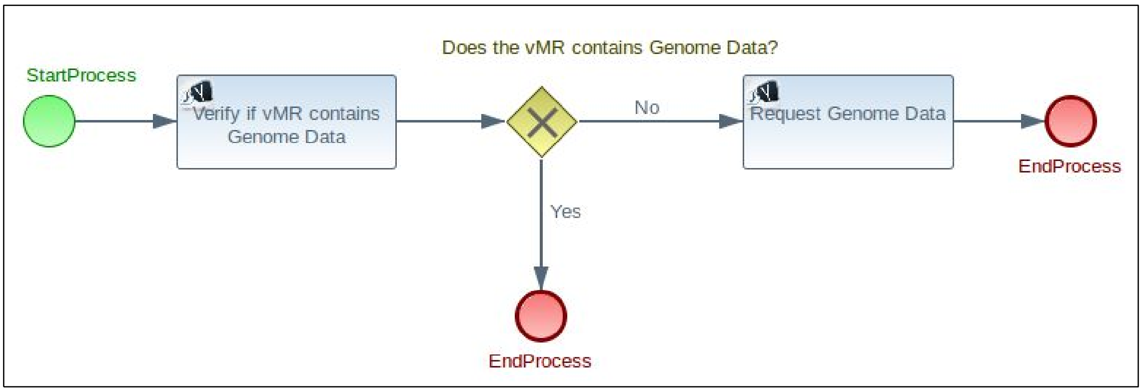


Figure 23: BPMN process that defines the evaluation logic of the vMR.

4.8. Clinical Decision Support Service

The CDS service is provided by the open source system OpenCDS, which is an implementation of the DSS standard. Figure 24 shows how service clients (e.g., the ESB) interact with the CDS service interface.

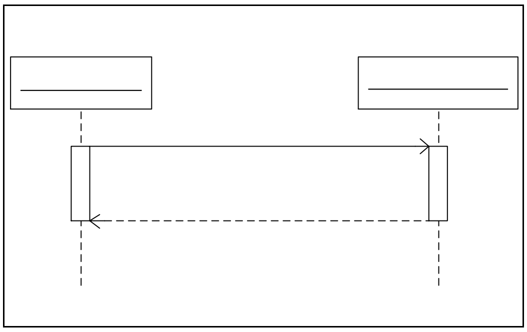


Figure 24: Interaction between CDS clients and the CDS provider.

An example of a SOAP CDS service request is shown in figure 25. The element on line 9 (*kmEvaluationRequest*) indicates the CDS knowledge module to use to evaluate the request and the line 17 (*base64EncodedPayload*) contains the vMR document encoded in Base64 format.

```xml
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header/>
  <env:Body>
    <ns2:evaluateAtSpecifiedTime xmlns:ns2="http://www.omg.org/spec/CDSS/201105/dss">
      <interactionId interactionId="InteractionId-1754138899" scopingEntityId="edu.utah.bmi" submissionTime="2012-11-14T17:49:32.567Z"/>
      <specifiedTime>2012-11-14T17:49:32.568Z</specifiedTime>
      <evaluationRequest clientLanguage="" clientTimeZoneOffset="">
        <kmEvaluationRequest>
          <kmId businessId="WGS_v1" scopingEntityId="org.opencds" version="latest"/>
        </kmEvaluationRequest>
        <dataRequirementItemData>
          <driId itemId="testPayload">
            <containingEntityId businessId="123.456.7.8.2.1" scopingEntityId="edu.utah.bmi" version="1.0.0"/>
          </driId>
          <data>
            <informationModelSSId businessId="VMR" scopingEntityId="org.opencds.vmr" version="1.0"/>
            <base64EncodedPayload>
PD94bWwgdmVyc2lvbj0iMS4wliBlbmNvZGluZz0iVVRGLTgilHN0YW5kYWxvbmU9Inllcyl/Pg0KPG5zMjpjZHNJbnB1dCB4bWxuczpuczI9Im9yZy5vcG
VuY2RzLnZtci52MV8wLnNjaGVtYS5jZHNpbnB1dCI+DQoJPHRlbXBsYXRlSWQgcm9vdD0iMi4xNi44NDAuMS4xMTM4ODMuMy4xODI5LjExLjEuMS4
xli8+DQoJPHZtcklucHV0Pg0KCQk8dGVtcGxhdGVJZCByb290PSIyLjE2Ljg0MC4xLjExMzg4My4zLjE4MjkuMTEuMS4yLjEiLz4NCgkJPHBhdGllbnQ
+DQoJCQk8aWQgcm9vdD0iMS4yLjEuMTIwMDAiIGV4dGVuc2lvbj0iMzMzMzMiLz4NCgkJCTxkZW1vZ3JhcGhpY3M+DQoJCQkJPGJpcnRoVGltZSB
2YWx1ZT0iMTk0NTAyMTMiLz4NCgkJCQk8Z2VuZGVyIGNvZGU9Ik1hbGUiIGNvZGVTeXN0ZW09IjIuMTYuODQwLjEuMTEzODgzLjUiIGNvZGVTeXN0Z
W1OYW1lPSJITDciIGRpc3BsYXlOYW1lPSJNYWxlli8+DQoJCQk8L2RlbW9ncmFwaGljcz4NCgkJCTxjbGluaWNhbFN0YXRlbWVudHM+DQoJCQkJP
HByb2JsZW1zPg0KCQkJCQk8cHJvYmxlbT4NCgkJCQkJCTx0ZW1wbGF0ZUlkIHJvb3Q9IjIuMTYuODQwLjEuMTEzODgzLjMuNzk1liBleHRlbnNpb249I
ilvPg0KCQkJCQkJPGlkIHJvb3Q9IjEuMi4xLjEyMDAwliBleHRlbnNpb249ImVjaHl6cGF0aWVudC05MjYwOTQ6cHJvYmxlbXM6YWN0aXZlLTkyNjEwOSI
vPg0KCQkJCQkJPGRhdGFTb3VyY2VUeXBlIGNvZGU9IkNsaW5pY2FsliBjb2RlU3lzdGVtPSIyLjE2Ljg0MC4xLjExMzg4My42Ljk2liBjb2RlU3lzdGVtTm
FtZT0iU05PTUVELUNUliBkaXNwbGF5TmFtZT0iliBvcmlnaW5hbFRleHQ9lilvPg0KCQkJCQkJPHByb2JsZW1Db2RlIGNvZGU9IjMxNTA1ODAwNSIgY29
kZVN5c3RlbT0iMi4xNi44NDAuMS4xMTM4ODMuNi45NilgY29kZVN5c3RlbU5hbWU9IlNOT01FRC1DVClgZGlzcGxheU5hbWU9IkhlcmVkaXRhcnkgbm
9ucG9seXBvc2lzIGNvbG9uIGNhbmNlciAoZGlzb3JkZXIpli8+DQoJCQkJCQk8cHJvYmxlbUVmZmVjdGl2ZVRpbWUgbG93PSIyMDE0MDIxMylgaGlnaD
0iMjAxNDAyMTMiLz4NCgkJCQkJPC9wcm9ibGVtPg0KCQkJCTwvcHJvYmxlbXM+DQoJCQkJPHByb2NlZHVyZUV2ZW50cy8+DQoJCQk8L2NsaW5p
Y2FsU3RhdGVtZW50cz4NCgkJPC9wYXRpZW50Pg0KCTwvdm1ySW5wdXQ+DQo8L25zMjpjZHNJbnB1dD4=</base64EncodedPayload>
          </data>
        </dataRequirementItemData>
      </evaluationRequest>
    </ns2:evaluateAtSpecifiedTime>
  </env:Body>
</env:Envelope>
```


Figure 25: Example of a SOAP CDS message request.

4.8.1. CDS Service evaluation logic

OpenCDS uses the open source Drools rule engine to define the logic for evaluation of CDS requests. The knowledge module used to evaluate the genome data consists of four rules, which are shown in figure 26. This is a screenshot of the Drools Guvnor web application that allows editing and managing rules based on the Drools engine.

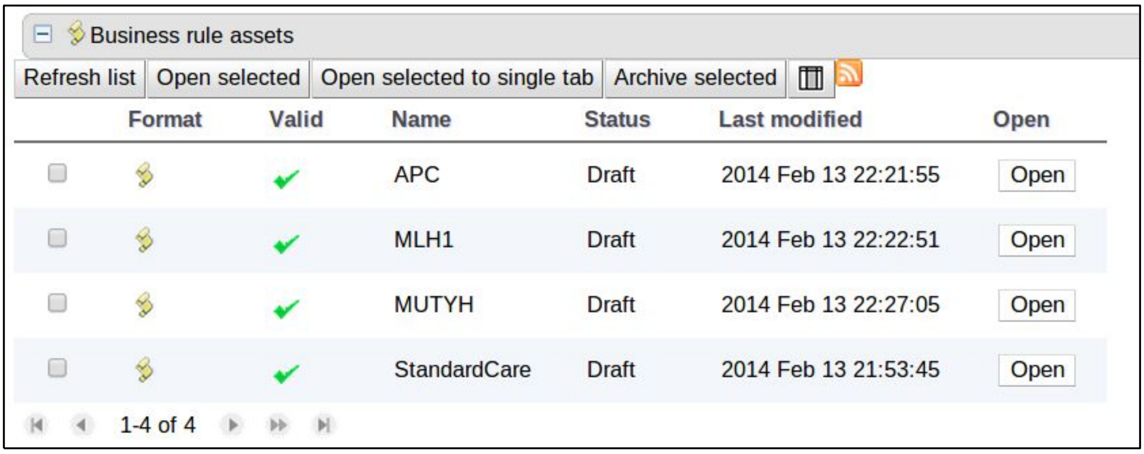


Figure 26: Knowledge based rules used to evaluate genome data.

Figure 27, shows the MUTYN rule expressed as Domain Specific Language (DSL) on the Drools Guvnor interface. Basically, this rule recommends colonoscopy screening for male patients between 20 and 85 years old, looking for pathogenic mutation of the gene MUTYH that has not received colonoscopy screening in the last 6 months.

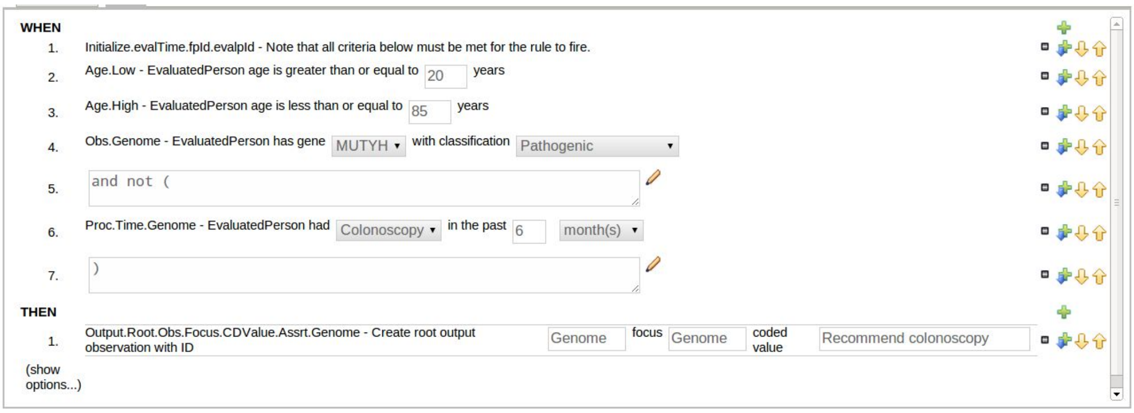


Figure 27: MUTYN rule.

4.9. EHR Interfacing Plugin

The core capabilities provided by CDS at the point of care is to improve clinical outcome and efficiency. These capabilities are realized when CDS functionality is integrated into the routine EHR clinical workflow [117] [118]. For this reason an EHR system was integrated into the architecture in order to verify the functionality for a complete patient-physician encounter (i.e. the time when the physician enters patient information and after processing by the rules engine, the CDS functionality is manifested on the EHR screen). The EHR system selected is the open source healthcare information platform Tolven, which is certified by the Office of the National Coordinator (ONC). Tolven is a meaningful certified EHR application. The meaningful use stage 2 core measures states "Use clinical decision support to improve performance on high-priority health conditions". Standards based CDS functionality in Tolven EHR necessitates the incorporation of the DSS standard. To accomplish this goal, a DSS interface plugin was developed for Tolven. The figure 28 shows an overview of this Tolven plugin and CDS service.

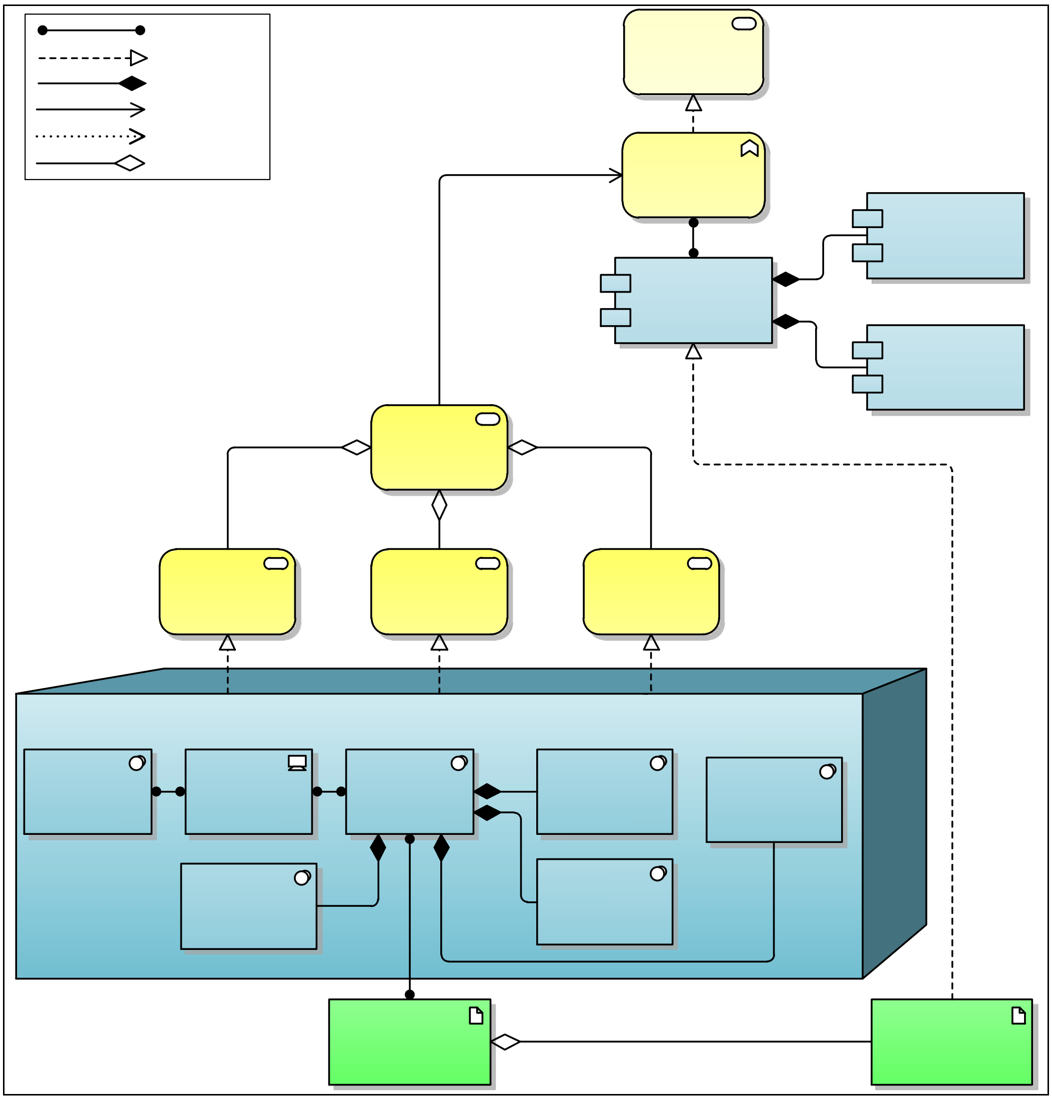


Figure 28: Tolven platform and CDS Service plugin overview.

As is shown in figure 28, the Tolven platform was installed in an x64 computer running a Linux CentOS version 5.8. The Infrastructure Services provided by the Tolven platform include:

a. Rule Engine,
b. Java EE implementation
c. Access to Patient data (e.g. lab results, observations and procedures).

Table 18: Hardware used to deploy Tolven EHR.

Hardware	
Processor	Intel® Core™ i3 CPU 4160 @ 3.60GHz × 4, 64-bit
Hard drive	250 GB
Memory	16 GB

Table 19: Software used to deploy Tolven EHR.

Software	
Operational System	CentOS Version 5.8, 64-bit
JVM	Oracle JVM 1.6
EHR	Tolven version 2.1: • PostgreSQL 9.0 • JBoss Application Server 6.0 • JBoss Rule engine • OpenDS

Tolven provides these Infrastructure Services by combining capabilities from other open source applications such as PostgreSQL, OpenDS, JBoss Application server and Drools rule engine.

The CDS Service interface is enabled by the "*CDS Service Plugin*", which is represented as an artifact. This artifact realises the Application Component "*CDS Tolven Service*" which contains two main components "*cds.drl*" and "*OperationBean*". The Application Component "*cds.drl*" is a Drools file that contains rules that are triggered when the user (e.g. physician) enters new information into the patient record. These rules create an instance of the *OperationBean,* which along with other Java classes is responsible for collecting patient information, creating a vMR with the collected information and for sending the CDS Service request. The rules contained in the *cds.drl* file also specify to display the results of the CDS Service request in the Alerts section of the patient overview for the EHR interface. The figure 29 shows a screenshot of the Tolven user interface with the CDS service response displayed in the Alerts section.

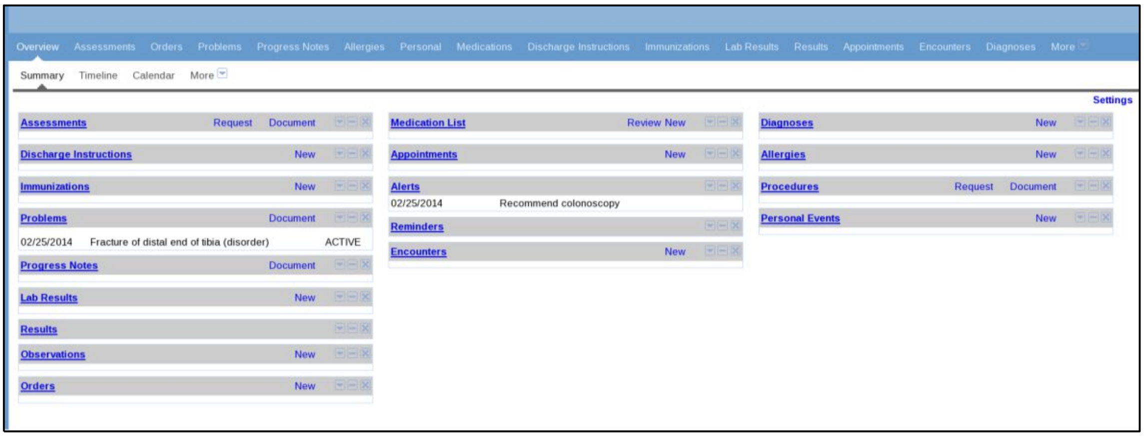


Figure 29: Tolven patient summary user interface showing CDS capability in Alerts section.

4.10. Performance Evaluation

The speed at which the CDS is delivered is an important factor for effective implementation [177]. Therefore, a performance test of every service and the overall architecture was conducted using the open source tool LoadUI (see table 20 for summary). The WGS Data service handled 3,109 requests over a five minute period. This service fastest request took 25 milliseconds (ms), and the slowest took 697 ms, with an average of 40 ms (SD 47.44 ms). OpenCDS handled 3,015 requests over a five minute period. This service fastest request took 7 ms, and the slowest took 914 ms, with an average of 12 ms (SD 17.04 ms). The ESB service, which includes service calls to the WGS data service and OpenCDS and closely represents the overall performance was also evaluated. Due to the hardware limitations of our machine, the test was limited to 20 simultaneous users. The ESB service handled 650 requests over a five minute period. The fastest request took 356 ms the slowest took 4,243 ms, with an average of 944 ms (SD 621.04). It is important to note that the average response time is less than one second.

Table 20: WGS enabled CDS service Performance test results.

Component	Simultaneous Users	Total requests handled	Min request time (time in ms)	Max request time (time in ms)	Average request time (time in ms)	Standard deviation
WGS Service	100	3109	25	697	**40**	47.77
OpenCDS	100	3015	7	914	**12**	17.04
ESB	20	650	356	4243	**944**	621.04

4.11. Results

1. ***Architectural Approach:*** This architectural approach based on SOA allows standards based CDS service enhanced by WGS information. This is an enriched knowledgebase for CDS for diagnosing and treating genetic conditions and can be reused and further extended for more complex conditions. The SCA composite can be reused for implementing automated CDS for other complex genetic conditions.
2. ***Early Diagnosis of colorectal cancer***: This system is able to request genetic information for adults at risk from the Lynch Syndrome condition. The CDS capability provides recommendations for performing colonoscopy for patients at

risk. If this testing is timely it will result in early diagnosis thus providing prompt treatment options for patients and prevent long term health deterioration.

3. ***Population Health Management***: This implementation provides functionality for screening adults with pathogenic gene variants for Lynch Syndrome. For larger populations, people with positive DNA screening with an age greater than 20 can be separated out form a population of patients registered at a primary care facility or hospital.
4. ***Standards based Solution***: The integration solution implements standards for SOA implementation like HL7 RLUS and HL7 DSS. These standards were developed by the Health Services specification Project (HSSP). The HSSP standards are specifically designed to ensure a SOA based interface for healthcare data retrieval and data processing. Hence this study is an example of a SOA approach for development in the healthcare industry.

4.12. Conclusions

WGS information can facilitate the implementation of personalized medicine and therefore improve health care. However, there are several barriers that prevent the use of WGS information in clinical settings. Some of these barriers include: 1) limited knowledge of genetics among physicians, 2) complexity of genome information, 3) and lack of experts in genetics in clinical environments. Delivering CDS services within the clinical workflow at the point of care and at the time of decision making though the EHR allows the integration of genetic information to improve health care service delivery. To evaluate this approach, a SOA-based architecture was developed and tested that enables CDS services based on WGS information. The results obtained from this case study demonstrates that this approach is feasible and valid. However, large-scale validation could be performed in the future in real clinical settings,

Chapter 5

A Service Oriented Approach for Guidelines-based Clinical Decision Support using BPMN and Business Rules

5.1. Introduction

The purpose of this chapter is to answer the research question: Is it possible to implement an SOA-based environment using CDS services built upon modelled clinical guidelines combining the business process language BPMN and business rules? Healthcare workflows have been modelled using different techniques and BPMN is a relatively new technology in this field. The following sections describe the architecture and implementation of a CDS service that besides meeting SOA requirements (e.g. loose coupling, reusability and composability) it allows implementation of clinical guidelines using the BPMN language and business rules. To demonstrate the functionality of the proposed architecture two CDS services based on clinical guidelines defined by the National Institute for Health and Care Excellence (NICE) [178] have been implemented: 1) preliminary diagnosis for Chronic Obstructive Pulmonary Disease (COPD), and 2) preliminary diagnosis for Lung Cancer.

5.2. Selection of NICE guidelines

NICE is the most important organization in the field of clinical guidelines development in the United Kingdom. NICE is an independent organization funded by the government, which provides guidance to the National Health Service (NHS). NICE is internationally recognised for the rigorous process of guidelines development [179], that is based on the best available evidence and also advice from experts, patients and industry. In order to create guidelines, NICE relies on external centres such as the National Clinical Guideline Centre (NCGC) [180], the National Collaborating Centre for Cancer (NCCC) [181], the National Collaborating Centre for Women's and Children's Health (NCC-WCH) [182], and the National Collaborating Centre for Mental Health (NCCMH) [183]. The NICE's guidelines are continually updated to keep pace with the continuous advances in medicine. Each clinical guideline is reviewed every two years to decide whether an upgrade is required [184]. Nevertheless, clinical guidelines can be updated more often as needed. The use of NICE's guidelines facilitates healthcare professionals in applying evidence based medicine (EBM) for the benefit of patients. EBM is the "conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients" [185]. Despite of the benefit of clinical guidelines, their implementation in practice is not always successful due to factors such as complexity of the guidelines [186].

To accelerate the implementation of guidelines in practice is often necessary to adopt various strategies such as combining guidelines with educational interventions, implement audits and other actions of improvement [186]. An action that has been proven effective is the delivery of recommendations at the point of care through the EHR interface using CDS modules [36]. Often these CDS modules are tightly coupled with the EHR, which prevent sharing CDS knowledge with other systems [36]. Furthermore, these CDS modules are usually developed using specialized clinical guideline modelling languages, which has proven to be slow and arduous [187]. In the following sections a new approach for developing CDS modules is proposed. This approach facilitates modelling of clinical guidelines using the business process language BPMN along with business rules. Additionally, the proposed architecture is based on the SOA principles, which enable sharing CDS knowledge with multiple systems.

5.3. Development Methodology

The Agile Business Development Methodology (ABRD) [188] was adopted as strategy to define the rules and business process model for both CDS services. The iterative method consists of six steps (see figure 30) that ensure that the CDS knowledge is encapsulated in the rules and the BPMN process can be easily updated to meet the evolving nature of the clinical guidelines. Based on ABRD the following steps were followed:

- *Harvesting*: Identify the rules as reusable CDS knowledge components for a clinical guideline.
- *Prototyping*: Design a model to represent the rules as part of a clinical process and validate the rules against the business logic they represent.
- *Building*: Build executable rules and deploy the rules in a runtime environment, and expose them as Web services to be consumed by requesting applications.
- *Enhancing*: Follow an iterative approach to modify existing rules and integrate changes as they appear in a clinical scenario.

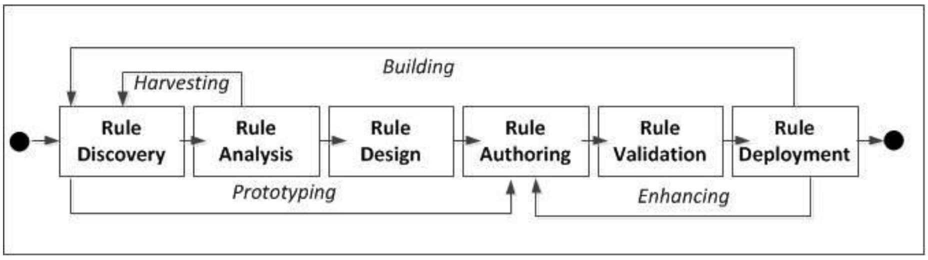


Figure 30: Agile Business Rules Development Methodology [188] .

5.4. COPD Clinical Guideline

COPD is a general term used to describe a number of conditions including chronic bronchitis, emphysema, chronic obstructive airways disease and chronic airflow limitation. This chronic disorder prevails with age and is usually associated with smoking. NICE provides evidence-based guidance on the most effective ways to prevent, diagnose and treat a myriad of conditions including COPD. NICE clinical guidelines are accessible in various formats including: narratives and summaries expressed as pathways. Figure 31 shows the NICE pathway for the COPD overview.

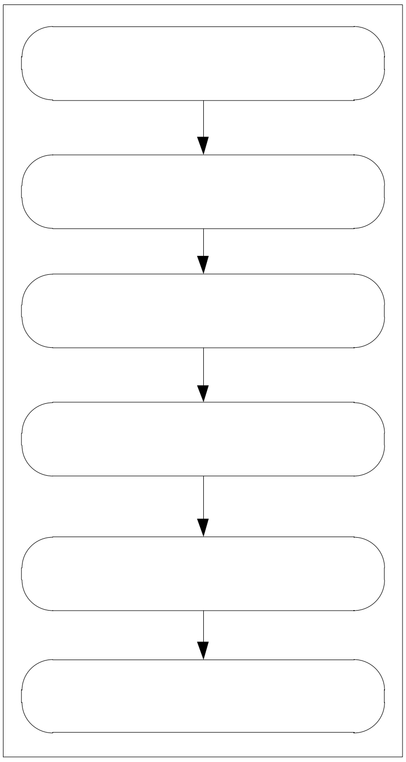


Figure 31: Adapted from the NICE Pathway for COPD overview [189].

The NICE COPD clinical guideline is quite extensive so for purposes of demonstration of the functionality of the proposed architecture only the task for preliminary diagnosis was implemented. The next section describes how the process of preliminary diagnosis was expressed combining rules and BPMN.

5.5. BPMN model for Preliminary Diagnosis of COPD

Identify the business process workflow and model the process visually were the first steps taken to describe the preliminary diagnosis of COPD using the BPMN language. The logic contained in each of the tasks defined in the business process model has been extracted and presented as business rules. This separation allows rapid changes in the logic of tasks such as age or assessment evaluation without modifying the business process model. As shown in figure 32, the first task of the process is to verify patient age. If the patient is older than 35 years, the process looks for symptoms of chronic cough, winter bronchitis, wheeze, exertional breathlessness, and sputum production. Note that the search for symptoms is carried out in parallel, as indicated for the parallel gateway (diamond with a plus sign internal maker). The process will look for the value of assessments, if the patient has any of these symptoms. One of the key requirements of this guideline is smoking assessment of the patient, which is recorded in the EHR as a preliminary step in evaluating the patient. This functionality is modelled and implemented in the workflow and reflected in the BPMN Process. Finally, if it is confirmed that the patient is a current smoker or former smoker, then the process will return as a result the recommendation "*consider a diagnosis of COPD*".

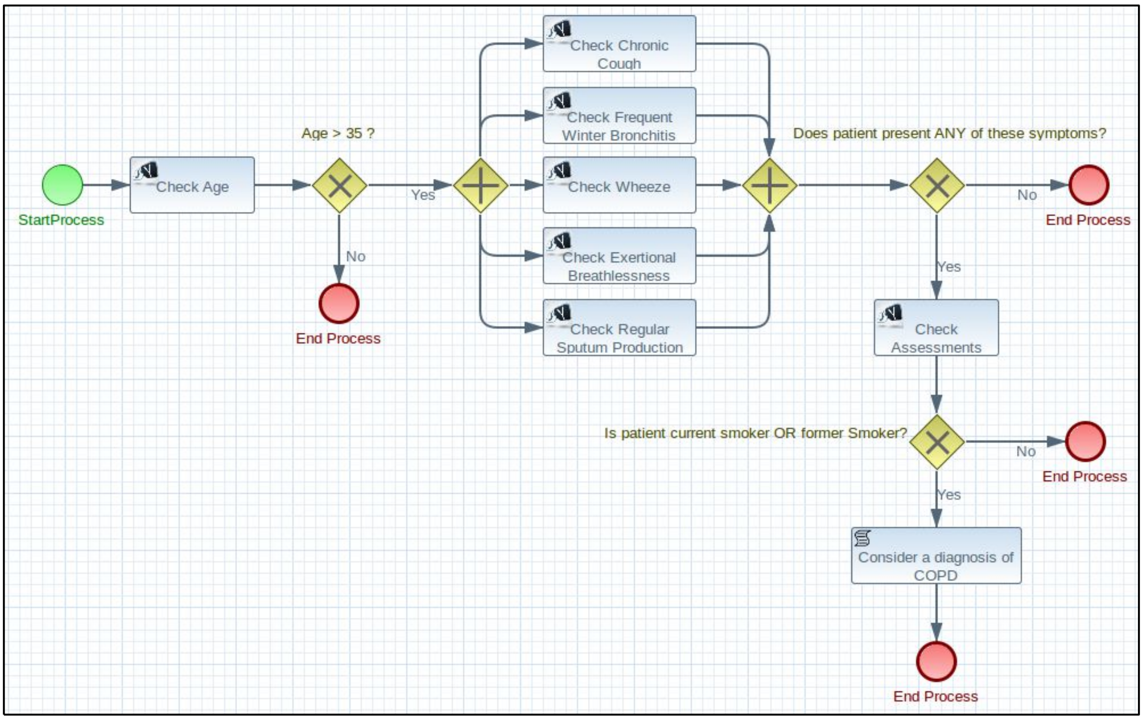


Figure 32: BPMN model for preliminary COPD diagnosis.

The logic contained in each of the tasks of the process is expressed as business rules. This allows the domain expert (i.e., physician or CDS expert) to make changes without having to modify the BPMN process. For example, the logic for the age evaluation can be modified by simply changing the value "35", as shown in figure 33. Every rule returns a Boolean value, which is then evaluated in the process model.

```
rule "CheckAge"
   when

        CDSInput(vmrInput.patient.demographics.age.value >= 35)

    then

        globals.put("Result", Boolean.TRUE);
end
```

Figure 33: Business rule for age evaluation.

It is worth noting that the rule condition (i.e., the when part or left side) is evaluating patient information contained in a data model, which is based on the HL7 vMR standard. This allows domain experts familiar with the XML vMR schema to define rules more easily. Most of the XML editors allow viewing the structure of XML schemas visually. Figure 34 is a screenshot of the XML editor Oxygen [190] that shows where in the vMR schema, patient age information is located.

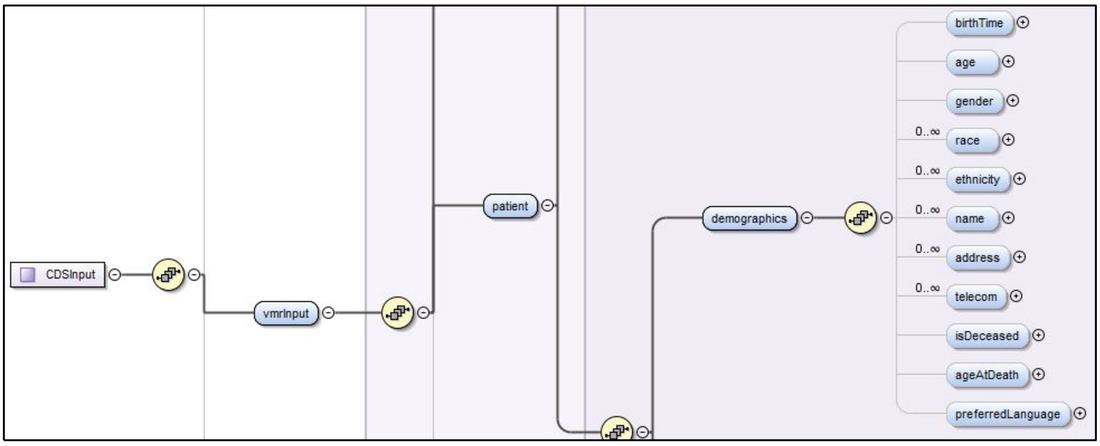


Figure 34: Snippet of the vMR schema using the XML editor Oxygen.

The evaluation logic of the rest of rules uses the same terminology system adopted by the EHR Tolven. For example to check if the patient presents “breathlessness” the business rule searches for the code “2070570062”, which is the SNOMED-CT code used to represent this this symptom in the EHR Tolven. The complete list of codes used in the present implementation is described in table 21.

Table 21: Terminology codes used on the CDS service for preliminary diagnosis of COPD.

Description	Terminology System	Code
Former smoker	UMLS	2007AA -3
Current every day smoker	UMLS	2007AA -1
Breathlessness	SNOMED-CT	207057006
Chronic cough	SNOMED-CT	68154008
Abnormal sputum	SNOMED-CT	274708000
Chronic obstructive bronchitis	SNOMED-CT	185086009
Wheezing	SNOMED-CT	56018004

The next section describes the CDS Service architecture and the components that provide the resources to deploy the BPMN model.

5.6. COPD CDS Service Component Architecture

To build an SOA-based service that allows deployment of the BPMN models such as described in the previous section, the architecture of the CDS system besides following the SOA principles has to contain a BPM engine, a rule engine and an interface that could enable other systems to consume the service. The proposed CDS service was built upon the SCA standard which facilitates the implementation of SOA principles. The open source infrastructure JBoss SwitchYard has been used as an SOA environment for the deployment of the SCA composite (figure 35). This infrastructure provides all the required components such as SCA engine, BPM engine, rule engine and message transformation.

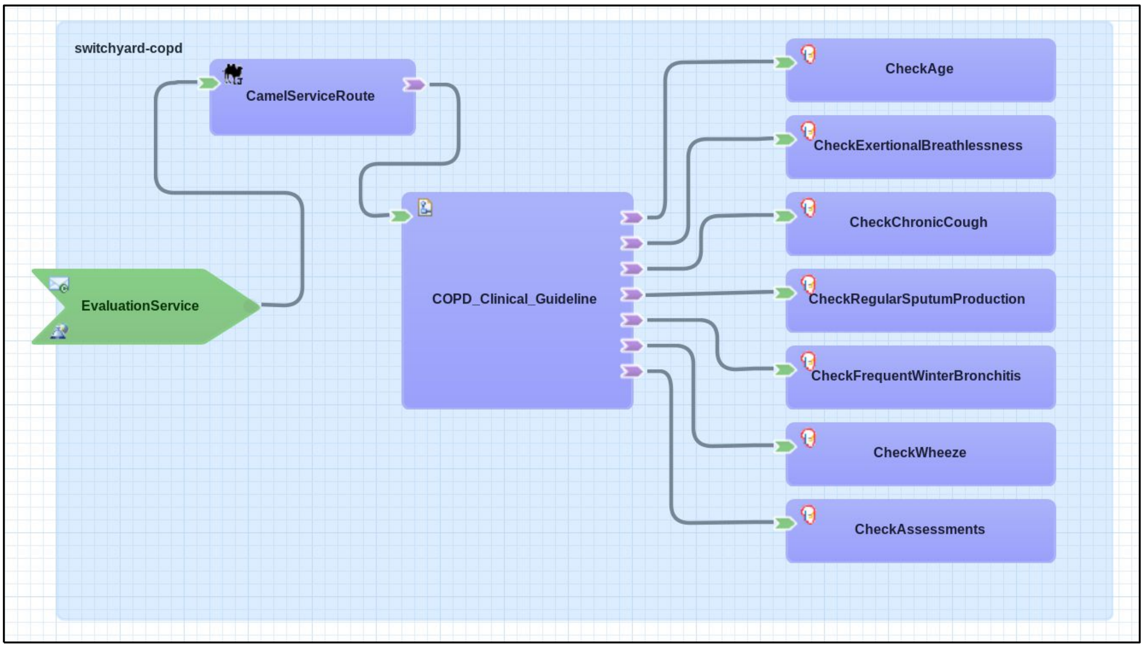


Figure 35: SCA composite that provides CDS service for preliminary COPD diagnosis.

The CDS is exposed as a SOAP service (*EvaluationService*) which is based on the DSS standard. The *CamelServiceRoute* component is responsible for transforming the XML vMR contained in the SOAP request into a Java object. The *COPD_Clinical_Guideline* component contains the BPMN model described in section 5.5 and is instantiated when a new Java object is received from the *CamelServiceRoute* component. The other SCA component wired to the *COPD_Clinical_Guideline* component contains the rules for each of the tasks defined in the BPMN process. The results of the BPMN process is converted to XML by the *CamelServiceRoute* component and returned as a CDS Service response.

The Tolven EHR plugin described in section 4.8 was used to demonstrate how the end user would receive the CDS service. Figure 36 is a screenshot of the Tolven EHR user interface showing the CDS as an alert.

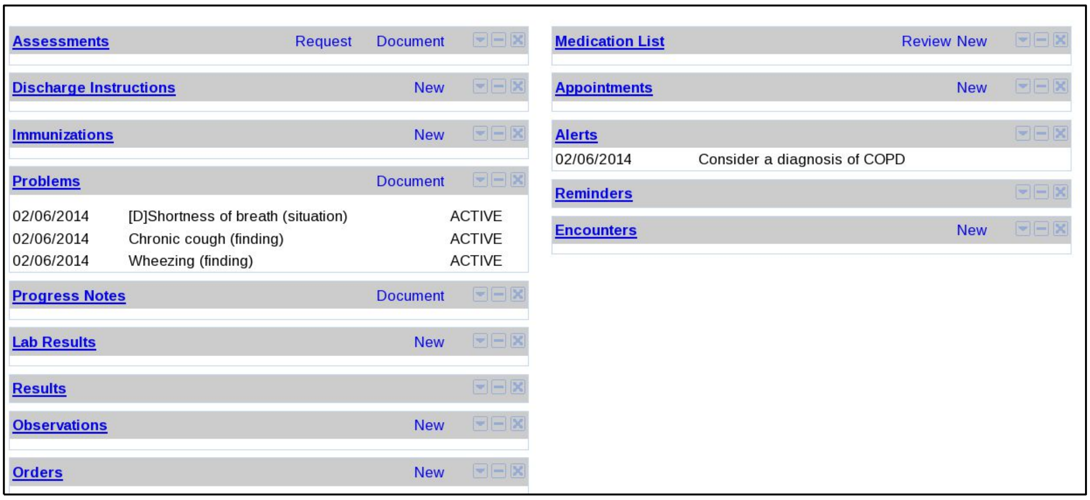


Figure 36: Tolven EHR user interface showing the CDS for COPD as an alert.

5.7. Lung Cancer Clinical Guideline

Lung cancer is characterized by the uncontrolled proliferation of lung cells, some of the symptoms include persistent cough, coughing up blood and shortness of breath. The NICE's clinical guideline CG121 [191] describes a sequence of actions that the physician must perform in order to consider a diagnosis of lung cancer. Figure 37 shows the pathway for lung cancer overview defined by NICE [192].

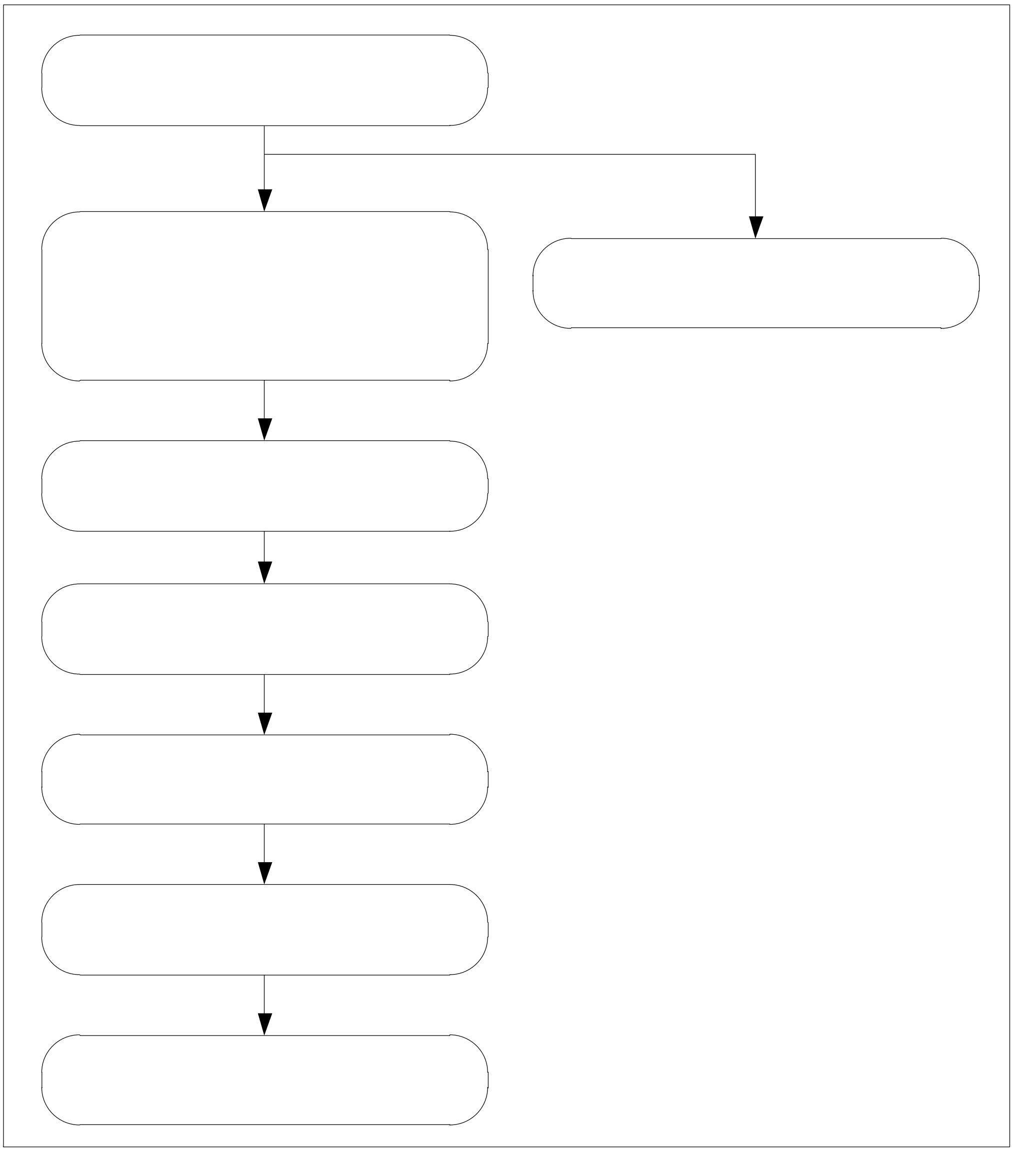


Figure 37: Adapted from the NICE Pathway for lung cancer overview [192].

5.8. BPMN model for Preliminary Diagnosis of Lung Cancer

Following the same approach described on section 5.2, a BPMN model combined with rules was defined in order to build a CDS service for the preliminary diagnosis of lung cancer based on the NICE guideline CG121. The business rules were used to identify the symptoms and the BPMN model describes the sequence of steps of the clinical pathway. Figure 38 shows the BPMN representation of this guideline.

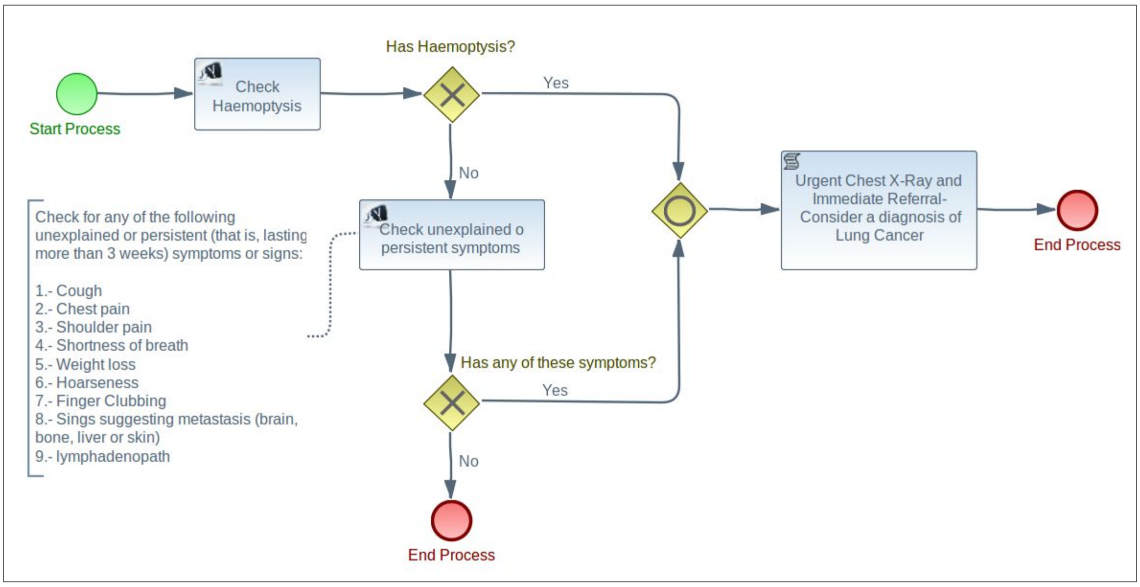


Figure 38: BPMN model for diagnosis of lung cancer.

The first task of the process is to check whether the patient has Haemoptysis. This verification is performed by a business rule which will return the Boolean value true if is this symptom is detected in the vMR. As shown in figure 39, the business rule searches for the value “66857006” that is the SNOMED-CT code for the haemoptysis disorder.

```
rule "CheckHaemoptysis"
    when
        $cdsInput : CDSInput(vmrInput.patient.clinicalStatements.problems.($problemList : problem ))
        $problem : Problem(problemCode.code ==  "315058005", $pname : problemCode.displayName) from $problemList
    then
        System.out.println("Problem identified : "  + $pname);
        globals.put("Result", Boolean.TRUE);
end
```

Figure 39: Business rule used to check Haemoptysis.

The complete list of codes that defined the symptoms is shown in table 22.

Table 22: Terminology codes used in the CDS service for preliminary diagnosis of lung cancer.

Description	Terminology system	Code
Haemoptysis	SNOMED-CT	66857006
Cough	SNOMED-CT	49727002
Chest pain	SNOMED-CT	29857009
Shoulder pain	SNOMED-CT	45326000
Shortness of breath	SNOMED-CT	207057006
Weight loss	SNOMED-CT	89362005
Hoarseness	SNOMED-CT	50219008
Finger clubbing	SNOMED-CT	30760008
Lymphadenopathy	SNOMED-CT	30746006

Similar to section 5.6, the CDS service is based on a SCA composite (see figure 40). The BPMN process is contained in the SCA component (*NICE_Lung_Cancer_Clinical_Guideline*) and the evaluation of symptoms is contained in two SCA components: *CheckHempoptysis* and *CheckUnexplainedSymptoms*. The message transformation is performed by the *CamelServiceRoute* SCA component and the CDS service is exposed as CDS Web service (green arrow, *Evaluation*).

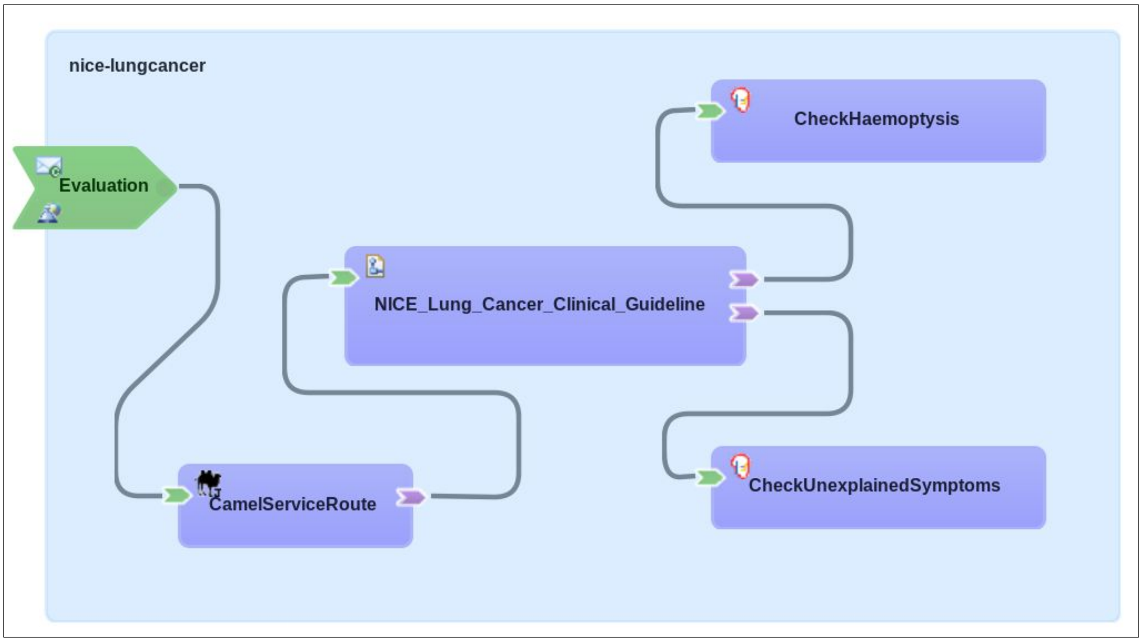


Figure 40: SCA Composite for lung cancer guideline.

Figure 41 shows the EHR user interfaces displaying the CDS provided by this service displayed as an alert.

Figure 41: Tolven EHR user interface showing the CDS for Lung Cancer as Alerts.

5.9. Results and Performance Evaluation

One of the advantages of using Web services as a means for delivering CDS is the ability to share CDS with multiple systems. Thus, to get an idea of how the services would behave when multiple systems send requests simultaneously, a load test was conducted using the open source application LoadUI. The LoadUI test strategy used was the "usage component", which allows simulating a number of systems making use of the service at the same time. Both CDS services were deployed individually in a computer with processor Intel Core i7, memory 16GB and running Linux Ubuntu 12.04 LTS 64-bit. The LoadUI tool was deployed in a separate computer with a processor Intel Duo core, memory 4GB and running Windows 7 64-Bit.

5.9.1. COPD CDS service performance test results

The load test for the COPD CDS service was carried out during 5 minutes simulating 20, 30 and 40 users. An overview of the test results is shown in table 23.

Table 23: COPD CDS service LoadUI test summary.

Test scenario	Simultaneous Users	Total requests handled	Min request time (time in ms)	Max request time (time in ms)	Average request time (time in ms)	Standard deviation
Test 1	20	604	200	1119	**293**	136.87
Test 2	40	1202	199	5354	**467**	473.33
Test 3	50	1541	198	4027	**719**	569.20

5.9.2. Lung Cancer CDS service performance test results

The load test for the Lung Cancer CDS service was also carried out during 5 minutes simulating 30, 50 and 60 users. The summary of the load test is shown in table 24.

Table 24: Lung Cancer CDS service LoadUI test summary.

Test scenario	Simultaneous Users	Total requests handled	Min request time (time in ms)	Max request time (time in ms)	Average request time (time in ms)	Standard deviation
Test 1	30	881	164	1524	**274**	162.13
Test 2	50	1504	163	4328	**567**	480.79
Test 3	60	1779	162	7290	**987**	883.37

In both tests some service responses took much longer than the average, however no fault was reported. It can be speculated that this behaviour is due to limitations in the hardware resources of the host machine. Additionally, it is noticed that the service response of the Lung Cancer CDS service is faster than the response of the CDS COPD service. This is because the process model is simpler and the business rules are contained in only two SCA components instead of one SCA component for each rule.

5.10. Conclusion

This chapter describes the design and implementation of CDS services for two case studies: 1) CDS service for the preliminary diagnosis of COPD, and 2) CDS service for preliminary diagnosis of Lung Cancer. Both services were developed based on clinical guidelines established by NICE. The clinical guidelines were modelled combining BPMN language and business rules. SCA technology was used to connect the various components and to expose the clinical guideline as Web service. This demonstrates the successful implementation of a service-oriented approach for guidelines-based clinical decision support combining the business process language BPMN and business rules. Both CDS implementations are based on relatively simple clinical guidelines. However, as the SCA standard fully supports the reusability aspect, these CDS services can easily be used by more complex SCA composites. Note that these CDS implementations are using the same terminology codes used by the EHR client (e.g., Tolven). This can limit the interoperability with other EHRs that would use different terminology systems. However, because of the flexibility provided by the SCA standard, it is possible to build a SCA composite that provides terminology mapping services to other components, so that the logic of the CDS service is not tied to a specific terminology system.

The development process of CDS services using the proposed approach can be summarized in the following steps: 1) Identify business process workflow – clinical pathway; 2) Model process visually – build the BPMN model; 3) Develop runtime components – develop tasks defined in the workflow as SCA components; 4) Deploy runtime engine – deploy the SCA composite into the runtime environment; 5) Instantiate runtime instance – execute request and response process between SCA composite and EHR. The separation of the logic of the tasks from the process model provides a significant level of flexibility that facilitates making rapid changes without having to modify the process model. Using this approach can considerably simplify the development of CDS services. Additionally, CDS module developers can benefit from the continuous advances of these technologies as they are widely used in other industries such as finance and banking.

The architecture of the CDS service uses the HL7 vMR standard as the central data model and exposes its functionality as a standard HL7 DSS service. This significantly facilitates the integration with other healthcare systems. This was shown through the Tolven plugin, which displays the CDS recommendations to the end user (e.g., physician) in the form of alerts in the EHR user interface. Showing the CDS recommendations directly in the EHR user interface ensures that EBM is incorporated into the clinical workflow and prevents physicians interrupting their work in order to locate, read and process clinical guidelines. An important aspect of the approach shown in this chapter is the adoption of the systematic method ABRD for transforming clinical guidelines into business process and business rules. This ensures that the CDS knowledge contained in the rules and the process model can easily be updated to keep pace with changes in the clinical guidelines.

Chapter 6

Human Task Management

6.1. Introduction

This chapter evaluates the architecture approach developed and tested in Chapter 4 and 5 for Organizational processes. The previous chapters have tested this approach for Clinical Practices for Diagnosis and WGS. This chapter further extends the capabilities of this architecture solution. Healthcare processes require the synchronization of multidisciplinary resources and services in order to deliver effective clinical outcomes [193]. BPM technology can be used to coordinate communication between different actors, (e.g., computer and human) and automate different tasks. This can result in better patient care and ensure evidence based medical practice [194]. The Emergency Department (ED) is one significant example of human interaction workflows [195]. Humans are actively involved and continually interact with information systems in an ED. One of the serious problems encountered in an emergency department is Patient Overcrowding. To address this issue, it is necessary to maintain a rapid patient flow by accurately coordinating resources that are responsible for treating patients. This chapter describes a human task management architecture component that allows the coordination of tasks specified in workflows such as an ED. This component is based on BPM technology and has been incorporated into the architecture that has been described in chapters four and five, thus enabling additional functionality from an organizational point of view. The features of the human task management component are demonstrated by implementation of the UK NICE guideline for managing Acutely ill Patients in the ED in the following sections.

6.2. Support for Organizational Processes

In order to understand the potential benefits that healthcare organisations can derive from adoption of a SOA architecture, including human task management, it is important first, to distinguish between organizational process and medical processes [196]. Organizational processes establish the management and coordination among different departments as well as communication between different healthcare professionals across these departments or divisions. Examples of such processes include laboratory orders and operational management in a hospital. Electronic healthcare information exchange standards play an important role in organizational processes. On the other hand, clinical processes are related to diagnostic and treatment plans for a specific patient. These processes are driven by several factors such as ever-evolving medical knowledge, experience of the physician, clinical situations, clinical guidelines and pathways, to name a few. Owing to the complex nature of these processes, it is sometimes very difficult to automate them [195]. Studies have shown that providing medical knowledge intelligently filtered and delivered at the place and time of decision-making, through information systems such as EHR can improve clinical outcomes [36]. The architectural model described in chapters four and five focuses primarily on supporting clinical processes. However, there is a need for an architecture that can support processes that require human intervention to be completed (e.g., organizational processes). In order to support organizational processes the architecture has been extended by a adding a human task management component, which is described in the following sections.

6.3. Architecture Overview

In order to enable human interactions in the architecture described in chapter four and five, a human task management component is introduced. This component is based on the open source jBPM [197] that provides a comprehensive platform for business process management and it is based on the standards WS-HumanTask and BPMN. Following the SOA approach, this component has been deployed as an independent system, which can interact with the other architecture components (e.g., EHR and CDS service) via the ESB. The architecture can be divided into different layers, which are presented in the figure 42.

1. The layer at the lowest level is the Clinical Applications layer. It is composed of Clinical Information systems such as EHR, CDSS and LIS. These applications usually maintain their own database.
2. After the Clinical Application layer is the Enterprise Application Integration layer. This is composed of the Enterprise Service Bus addressing the integration issues. This is achieved by creating application adapters and implementing protocols for message routing and transformation.
3. Above the EAI layer is the Service Interface layer, which provides a bridge between EAI and the higher SCA runtime layer. In figure 42 it is assumed that LIS provides a service interface that can be used to interact with higher levels.
4. Service composition functionality is provided by the SCA runtime. SCA composite services can invoke clinical applications either using standardized service interfaces or through the ESB. The example shown in figure 42, shows a service composition strategy based on BPM technology as exemplified in section

3.8.1. Using BPM technology as a means for service composition allows modelling of processes established by the healthcare organization or by clinical pathways.

5. The human task management component is presented as the second layer. Human interaction workflows are carried out in the human task management component. The tasks of the human interaction workflow can be performed by humans or services provided by systems. The activities of the human interaction workflow can also be part of organizational. Composite services represented in the subsequent layer can interact with the human task management component through the REST API provided by jBPM. The human task management component layer shows an example where a human interaction workflow is part of the organizational process of an Emergency Department.
6. Finally the top layer represents a routine ED workflow, which is realized once the Human Task Management component is executed.

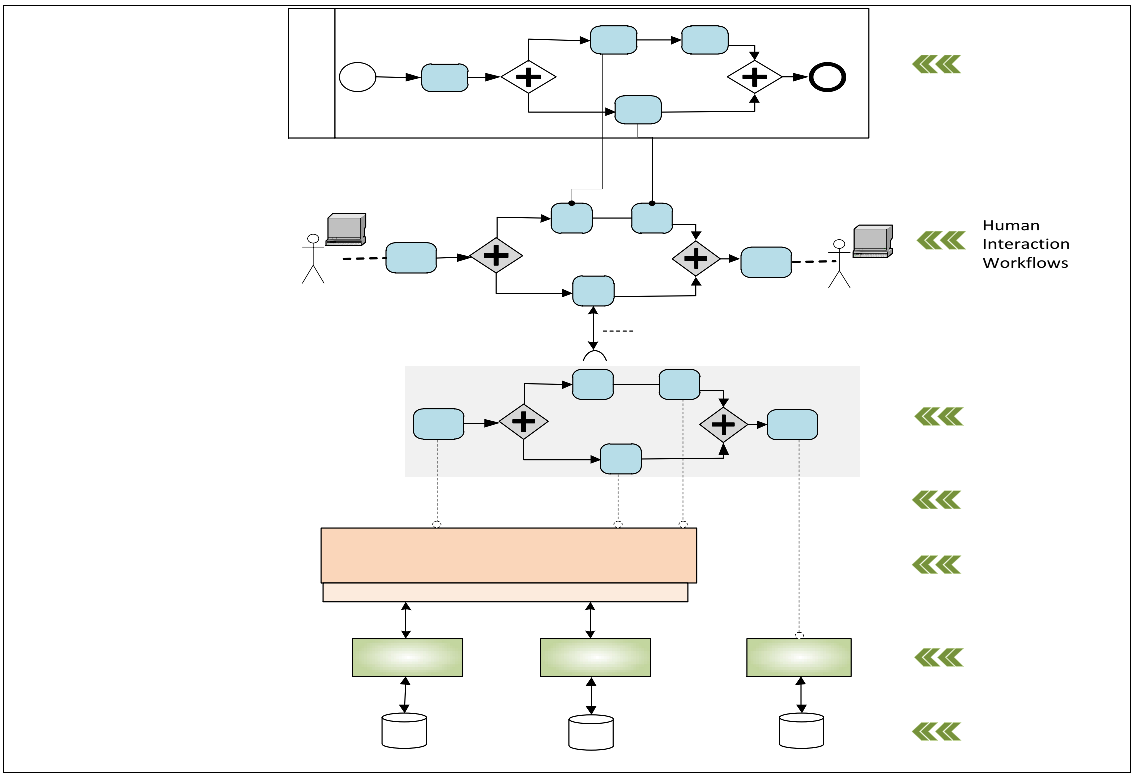


Figure 42: Architecture overview-SOA and BPM Components.

6.4. The Human Task Management Service deployment

Figure 43 shows the technical deployment of the human task management module. The application was deployed in an x64 computer running Ubuntu 12.04LTS. The Infrastructure Services JVM and Java EE Server are realized by Oracle JVM version 1.7 and JBoss Application version 7 respectively. These Infrastructure Services are used by the Application Component JBoss jBPM, which contains a business rules engine and a business process engine that supports the BPMN language. The Human Task Management function is assigned to JBoss jBPM, which realizes the Human Task Management Service. This service can be accessed via a web-based user interface or by the REST API. Processes can be initiated in two ways: 1) by a human actor using the jBPM web interface or 2) by the ESB using the jBPM REST API. The REST API provided by jBPM allows performance of tasks such as *start process, abort process* and *get process* status [197].

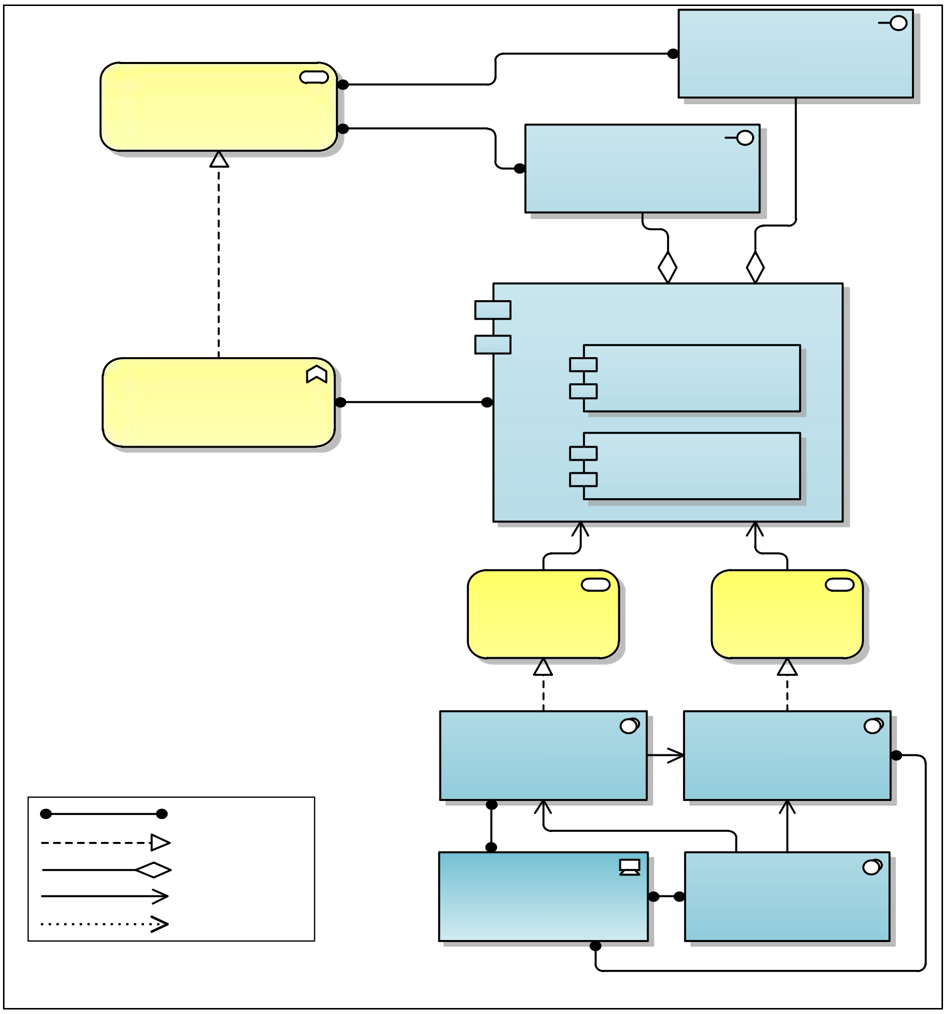


Figure 43: Overview of the Human Task Management Component.

6.5. Emergency Department Overview

Emergency Department (ED) also known as Accident and Emergency (A&E) assesses and treats serious conditions that require immediate attention and cannot wait for a normal appointment with a General Physician (GP). The conditions and requirements of ED are so specific that there was a need to create a field of medicine specifically to address such cases. This field is called Emergency Medicine with input from several organizations and research groups contributing continually to enrich the knowledgebase required. [198]. The key challenges that an emergency department is subjected to are, firstly, patient waiting time and secondly, length of stay during an ED encounter. These two issues result in overcrowding affecting the quality of care and worsening the conditions of critically ill patients [199]. These ED problems appear to be worsening in many other countries besides the UK [200]. Despite the importance of providing rapid medical care to critically ill patients, most Emergency Departments (EDs) fail constantly to admit patients within the time recommended by the prioritising process called triage [201]. While, there are many factors that contribute to this delay, studies have shown that improvements in ED processes such as triage, registration, discharge and allocation of resources can improve wait time and patient flow [202] [203]. Processes in ED require precise coordination of resources such as beds and equipment, technological resources and tasks such as admission, referral, and discharge in order to maintain a continuous and accurate patient flow. These resources are managed by a healthcare professional as well as support staff in the ED. For some of the routine processes, the care can be provided by ED Staff (nurse, physician, critical care team), whereas for some processes, information systems are responsible, (recording patient data such as labs, imaging etc.). It can be said that Emergency department care process is a combination of inputs from humans as well as computer based systems. This requires a mechanism for efficient collaboration between both types of actors.

6.6. NICE Guideline for Acutely ill Patients in the ED

The NICE guideline for acutely ill patients in the ED defines a standard workflow for managing critically ill adult patients in the ED [204]. The workflow assigns roles and responsibilities for the initial evaluation and the subsequent processing in case the patient health deteriorates. The tasks defined in the guideline are the responsibility of three main actors: 1) Admitting doctor, 2) Bedside nurse, and 3) Electronic Physiological Track and Trigger system (PTTS). The interaction between these actors defines the patient flow based on the patient status. The NICE guideline for acutely ill patients in the ED attempts to tackle some of the challenges present in ED such as reducing patient wait time and streamlining patient flow. The activities defined in this clinical guideline are described below:

Admitting Patient:

1. Receives the patient
2. Performs initial evaluation
3. Orders routine monitoring. In this stage the responsibility is transferred to the bed side nurse.

Bedside Nurse

1. Performs routine monitoring. This includes, vital signs, lab tests as ordered by the admitting doctor.
2. Records the results of the routine monitoring in the electronic system, PTTS.
3. Received the results from the PTTS after processing. As indicated by the results provided by PTTS, the patient is assigned a score. Based upon the score, the relevant ED Resource is contacted.

PTTS:

1. Takes as input, the results of the routine monitoring, entered by the bedside nurse.
2. Processes the input and provides results to the user.
3. These results are assigned a criticality score:
    a. *Low* – Alert nurse in charge and increase frequency of observations.
    b. *Medium* – Urgent call to primary care team responsible for the patient. At the same time call specialist responsible for acute illness (i.e., critical care outreach team, hospital-at-night team or specialist trainee in an acute medical or surgical speciality)
    c. *High* – Emergency call to the team responsible for critical care.

6.7. Implementation methodology

The implementation methodology "Pattern-Based Lifecycle For E-Workflow" [205] was adopted as the strategy to implement the UK NICE Guideline for Acutely ill Patients in the ED. This methodology ensures the integration of the following four organizational aspects into the design of the executable business process:

1. **Functional perspective**: Break down functional processes into tasks that are performed by humans or information systems.
2. **Organizational perspective**: Define the roles and actors involved in the process and specify how the organization manages its resources.
3. **Behavioural perspective**: Define how the information flows through the different steps of the workflow and specify the organization policies, rules and practices.
4. **Informational perspective**: Define the information that will be consumed or produced by the organization.

Three main actors were identified: Admission doctor, Bedside nurse and the Physiological Track and Trigger System (PTSS). These actors work in collaboration to provide care based on the severity of illness. The results of the analysis of the clinical guideline are summarized and categorized according to the implementation methodology in table 25.

Table 25: Summary of analysis of clinical guideline according to e-Workflow design life cycle.

Functional	Organizational	Behavioural	Informational
Admit Patient	Assign ED Resource	Contact ED Resources	Patient Information
Perform Assessment	Transfer Patient	Assign Group	Resource Information
Perform Monitoring	Ward Management	Contact Departments	PTTS Results

The clinical guideline was designed using the BPMN language on the Web console JBoss jBPM as shown in figure 44. The three process participants are arranged in swim lines. The process initiates with the patient admission to ED by the Admission doctor. Based on the results provided by the PTTS, the Bedside nurse will contact the appropriate resource. The PTTS assigns three critical patient scores: low, medium and high critical.

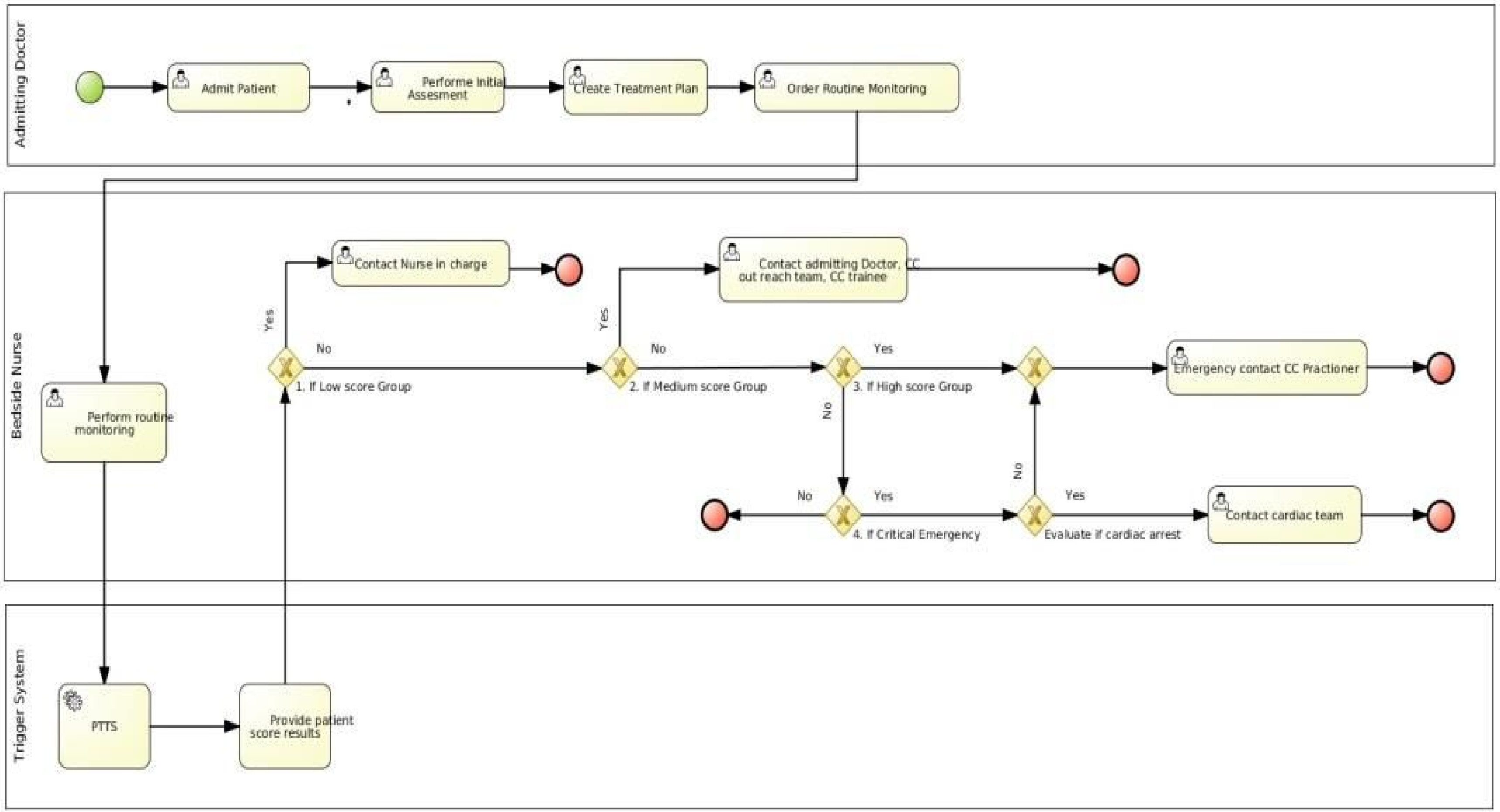


Figure 44: BPMN model of e-Workflow for ED Resources.

Having this model running on the human task management module allows users from different levels (i.e., physicians, nurses and administrator) to monitor the patient's condition. Based on the roles assigned to the different users in the hospital, the system offers different functionalities such as place order and claim task.

6.8. Conclusion

Healthcare organizations can benefit from IT architectures that allow modelling processes combining services provided by humans and information systems. Healthcare professionals and stakeholders can monitor the progress of a particular care process, ensuring delivery of evidence based medicine, track patient status and manage treatment resources. The patient can benefit from reduced waiting time and the allocation of the appropriate healthcare professional at the right time. The staff can receive accurate task assignments at the right time, contact the appropriate resources to assess the patient, monitor patient's condition and decrease treatment gaps. Adoption of service oriented standards such WS-HumanTask and BPM technologies such as the BPMN language promote interoperability and allow support for human collaboration scenarios (i.e., notifications to people involved, escalation and delegation of activities). This model for ED workflow will be tested in real clinical settings.

Chapter 7

Conclusions and Future Directions

7.1. Conclusions

7.1.1. Benefits of Using SOA based integration solution for CDS:

1. ***Orchestrating clinical tasks:***

   This SOA based solution for integrating healthcare applications is based on the principle of orchestrating different logic to allow a coordinated flow of information for the purpose of Clinical decision Support. The orchestration of the information required to provide CDS can be defined through a graphic human readable language that can be understood by domain experts and developers.

2. ***Interoperability:***

   The SOA based solution includes a service oriented middleware that supports healthcare messaging standards such as Decision Support Service and Retrieve, Locate and Update Service. Interface development for an EHR Application. Various components that collectively provide HIT for a healthcare service organization can be integrated using this solution. This includes legacy systems and previously used stand-alone applications.

3. ***Reusability***

   CDS services are structured as a set of reusable components that can be used in multiple CDS solutions. CDS services are exposed through standardized interfaces (e.g., HL7 DSS) which enable services to be reused by multiple healthcare systems.

4. ***Scalability:***

   This solution is built on well-defined and self-contained CDS services that are independent from other services; it allows the creation of a scalable architecture where new services instances can be added to fulfil additional demands. The SOA solution is based on a lightweight service oriented middleware that supports clustering which can be configured to provide high availability and load balancing.

5. ***Best Practice specification Implementation (service component architecture)***

   Service Component Architecture is the industry specification for implementing best practices for SOA based IT implementation for an enterprise. This research has leveraged this specification in the healthcare domain. Some of the benefits of using SCA are listed below:

a) Service composites are assembled into a single deployable artifact.
b) Service policies are independent of the programming code.
c) Services are independent of the programming language.
d) How the service is accessed is independent from the programming code.
e) Separation of business logic from details of its service implementation.
f) Ability to declare the quality of the service requirements, such as security, transactions and use of reliable messaging.
g) SCA Composites realize loose coupling between different modules.
h) The SCA Composites can be reused in more complex health related scenarios.

6. ***Enterprise Service Bus:***

ESB provides the foundations that realize the SOA runtime infrastructure where CDS services can be deployed. It allows management of communication across disparate health information systems. One of the key benefits of ESB is developing and integrating adapters for available healthcare applications, which results in an interoperable HIT implementation.

7. ***Business Rules and Business Process Model and Notation***

Clinical scenarios are implemented using BPMN models in combination with Business Rules. The use of BPMN in healthcare is a relatively new approach to provide a solution to improve clinical outcome and reduce medical errors.

8. ***Healthcare Enterprise Architecture:***

The resulting solution benefits the enterprise wide implementation of IT for healthcare industry by following a standards based implementation:

7.1.2. Standards Based Implementation:

This SOA based solution necessitates the implementation of standards in order to ensure an implementation of CDS capabilities at scale.

1. ***HL7 vMR:*** The standard vMR is been used as the common data model for the proposed SOA-based solution. Since the vMR has been designed specifically to

support CDS capabilities, the maintenance of CDS services is simplified as well as the transferring of clinical data between healthcare systems and the CDS services.

2. ***HL7 DSS:*** The DSS is the Web service standard adopted by the solution and it is used as a means to provide CDS. The DSS provides machine-interoperable, patient-specific assessments and recommendations based on the submitted clinical data. Conceptually, the DSS plays the role of a gatekeeper of one or more modules of clinical knowledge. The separation between the DSS and the modules of clinical knowledge allow adding new knowledge modules or modifying current knowledge modules without having to make changes in the code of the DSS implementation. The DSS adapter of the SOA-based solution only supports payloads encoded as vMR. However, this adapter can be extended to support different encoded payloads.
3. ***HL7 RLUS:*** The RLUS standard specifies a service-oriented interface that allows locating, accessing and updating clinical data regardless of the implementation data model of the healthcare system. The case study presented in chapter four "*Whole Genome Sequence Guided Clinical Decision Support*" demonstrates how RLUS facilitates the retrieval of patient genome data that is maintained in a separate system.

7.1.3. Software development approach:

1. ***Simplicity for Developers***

   It is based on a standardized data model (e.g., vMR). Developers can declaratively define basic service functionality such a service description, access methods and security without having to make changes to the code. Service components can be developed, debugged and tested separately, which promote reusability.
2. ***Architecture modelling using the Open Group ArhiMate language***

   Archimate is an international, open and vendor-independent standard. It focuses on the overview and coherence instead of the specificity and detail, as do other modelling languages such as UML and BPMN [143]. It allows the architect to accurately model, analyse, and visualize relationships between domains. It is supported by open source modelling tools.

3. ***Code Management/Version Management***

   GIT: Widely adopted open source distributed version control software.

   Maven: Widely adopted build automation tool. It simplifies the management of software dependencies.

7.1.4. Facilitating Domain Experts in Healthcare:

- ***Graphical Modelling using BPMN***:

  Healthcare Practitioners have the ability to participate in the Software development process by designing clinical practice models. Graphical modelling using BPMN allows clinicians to participate in the design process for HIT.

- ***Managing change***

  This architecture solution facilitates managing change to evolving Clinical knowledge. As new knowledge becomes available in the medical evidence, changes can be swiftly incorporated without the involvement of coding using programming languages.

- ***Developer-Physician collaboration***

  This solution has demonstrated minimization of the gap of understanding between: medical knowledge, software developers and physicians.

7.1.5. Clinical Practice Benefits:

1. ***Standards Based guidelines***

   Workflow automation is based on Standardised clinical practice guidelines provided by NICE UK. This enables the practice of evidence based medicine.

2. ***Clinical Guidelines Integration***

   The architecture solution allows the existing guidelines to integrate with more complex clinical scenarios, hence reusing existing knowledge.

3. ***Enabling Human Interaction:***

   This study has resulted in designing and implementing human interaction in an automated clinical workflow, hence allowing human and computer actors to conjointly provide service in the healthcare domain.

4. ***Early Diagnosis and Intervention:***

   The clinical case studies are focused on early diagnosis and intervention. The CDS service designed and implemented has been tested for performance.

5. ***Application to Personalized Medicine:***

   The WGS case study provides an implementation of using genetic data for clinical decision support. The availability of genetic data at the point of care has not been widely implemented in healthcare IT.

6. ***Organizational Practice:*** This research has resulted in providing a modelling solution for collaborative healthcare management. The different actors involved in managed care can benefit from this study to promote better coordination of tasks, activities and monitoring of events that require urgent attention from the management.

7. ***Auditable Processes:***

   BPMN has been widely used as an artefact to present an organization's processes for auditing for regulatory purposes. This research has focussed on representing clinical and non-clinical processes in a HCO using BPMN. These BPMN models can be used to present the HCO's functioning processes for the purpose of audit.

7.1.6. Benefits of Using Open Source Tools and technologies:

Having a platform based on open source technology and on open healthcare standards promotes sharing CDS implementation and best practices.

1. ***Promote sharable implementation:***

   For a healthcare enterprise, the use of open source tools and technologies allows development of sharable implementations of health-specific functionalities. The functionality code can be shared among different software developers' teams. The resulting code is reliable and robust as a number of developers have reviewed the code. Secondly, the open source tools and technologies and the resulting applications that are developed using them can be quickly deployed, hence saving time and resources resulting in decreased overhead costs. This is particularly important as open source tools do not generally include licensing costs.

2. ***Promote innovation***

   The open source development model promotes a high degree of innovation in an organization. Due to a collective participation in the development process it creates an environment of "collective invention". Mapping this idea to healthcare, the integration solution can serve as a platform for domain experts in CDS as well as developers to quickly diffuse innovative efforts into this solution.

   Secondly, the use of Open Standards promotes the adoption of these standards in a wider number of implementations.

7.2. Limitations

1. Hardware Limitations:

This architecture solution is based on integrating multiple components. Testing this solution requires hardware resources like multiple computers. This research was implemented using virtual environments hosted on a server machine. This infrastructure clearly needs more resources to improve the performance.

2. Limited availability of documentation for open source projects:

Since the open source tools and technologies are available as a result of online collaboration, documentation is not widely available.

7.3. Future Research

1. Terminology specification

A terminology service that facilitates multiple terminologies as specified by various EHRs will be developed.

2. SOA Governance:

This solution will be extended to include SOA governance concepts such as service registries and service discoverability. The S-RAMP [206] infrastructure will be explored for this purpose.

3. Cloud implementation:

The solution will be tested on the cloud to make the CDS service available for multiple healthcare settings.

4. Data store opportunities

More organizations are adopting applications that use semantic data to manage ontologies (e.g., *triple-stores*). Usually, these applications do not provide a service oriented interface. There is a need to enable such systems to participate in service oriented architecture. Some of the future efforts include the extension of the SOA-based solution in order to access multiple distributed ontologies for managing large streams of clinical data in healthcare settings.

5. Open Issues-:

Testing this solution with real-time patient data: The resulting solution from this research is currently being evaluated for load handling. In future, this solution will be installed in a real time healthcare setting to measure the outcome.

Implementing clinical guidelines involving multiple healthcare settings: The clinical guidelines that involve participation of multiple departments in a hospital will be tested for feasibility of this solution.

7.4. Final Comment

Distributed system implementations, as required in the healthcare industry, necessitate seamless integration of software systems that together result in improved outcomes. Service Oriented Architecture allows scaling up the integration of disparate applications and services in a platform-independent and interoperable manner. The combination of EA approaches like BPM and SOA together result in aligning the business processes (i.e. clinical and administrative processes) to meet the strategic goals of the healthcare enterprise. In addressing the complexity of systems integration, this research has shown that the SOA based framework facilitates robust connectivity for the multiple actors, including human actors, to provide Clinical Decision Support at the point of care, hence improving clinical care.

Bibliography

[1] Committee on Quality of Health Care in America and Institute of Medicine, *To err is human: building a safer health system*. Washington, D.C: National Academy Press, 2000.

[2] J. A. Osheroff, J. M. Teichc, B. Middletone, E. B. Steenf, A. Wrightg, D. E. Detmerh, J. M. Teich, B. Middleton, E. B. Steen, A. Wright, and D. E. Detmer, "A Roadmap for National Action on Clinical Decision Support," *J. Am. Med. informatics Assoc.*, vol. 14, no. 2, pp. 141–145, Jan. 2007.

[3] K. Kawamoto, C. a Houlihan, E. A. Balas, and D. F. Lobach, "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success.," *BMJ*, vol. 330, no. 7494, p. 765, Apr. 2005.

[4] A. Hevner and S. Chatterjee, *Design Resech in Information Systems*. New York, NY, USA: Springer, 2010.

[5] J. Estublier, "Software Configuration Management: A Roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, 2000, pp. 279–289.

[6] A. Arsanjani, "Service-oriented modeling and architecture," *IBM Dev. Work.*, no. January, pp. 1–15, 2004.

[7] P. Bianco, G. A. Lewis, P. Merson, and S. Simanta, "Architecting Service-Oriented Systems, TECHNICAL NOTE CMU/SEI-2011-TN-008 Research," 2011.

[8] A. Charfi and M. Mezini, "Hybrid Web Service Composition: Business Processes Meet Business Rules," in *Proceedings of the 2Nd International Conference on Service Oriented Computing*, 2004, pp. 30–38.

[9] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," *Computer (Long. Beach. Calif).*, vol. 40, no. 11, pp. 38–45, 2007.

[10] G. Baryannis, O. Danylevych, D. Karastoyanova, K. Kritikos, P. Leitner, F. Rosenberg, and B. Wetzstein, "Service Composition," in *Service Research Challenges and Solutions for the Future Internet SE - 3*, vol. 6500, M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger, Eds. Springer Berlin Heidelberg, 2010, pp. 55–84.

[11] C. Catley, D. C. Petriu, and M. Frize, "Software performance engineering of a Web service-based clinical decision support infrastructure," in *Softw. Eng. Notes (USA)*, 2004, vol. 29, no. 1, pp. 130 – 8.

[12] P. Ciccarese, E. Caffi, L. Boiocchi, S. Quaglini, and M. Stefanelli, "A Guideline Management System," *Medinfo*, pp. 28–32, 2004.

[13] D. Wang, M. Peleg, S. W. Tu, A. A. Boxwala, O. Ogunyemi, Q. Zeng, R. A. Greenes, V. L. Patel, and E. H. Shortliffe, "Design and implementation of the GLIF3 guideline execution engine," *J. Biomed. Inform.*, vol. 37, no. 5, pp. 305–318, 2004.

[14] P. Ram, D. Berg, S. Tu, G. Mansfield, Q. Ye, R. Abarbanel, and N. Beard, "Executing clinical practice guidelines using the SAGE execution engine.," in *Studies in health technology and informatics*, vol. 107, no. Pt 1, 2004, pp. 251–5.

[15] V. Bicer, O. Kilic, A. Dogac, and G. B. Laleci, "Archetype-Based Semantic Interoperability of Web Service Messages in the Health Care Domain," *Int'l J. Semant. Web Inf. Syst.*, vol. 1, no. December, pp. 1–22, 2005.

[16] K. M. Heard, C. Huang, L. a Noirot, R. M. Reichley, and T. C. Bailey, "Using BPEL to define an executable CDS rule process.," in *AMIA Annual Symposium Proceedings*, 2006, no. Mlm, p. 947.

[17] D. F. Lobach, K. Kawamoto, K. J. Anstrom, M. L. Russell, P. Woods, and D. Smith, "Development, Deployment and Usability of a Point-of-Care Decision Support System for Chronic Disease Management Using the Recently-Approved HL7 Decision Support Service Standard," in *Medinfo 2007: Proceedings of the 12th World Congress on Health*, 2007, vol. 129, pp. 861–865.

[18] M. M. Nakamura, W. W. Simons, R. Samuels, J. Daniel, and K. D. Mandl, "Service-oriented architecture for pediatric immunization decision support.," in *AMIA 2007 Symposium proceedings*, 2007, vol. 14, no. 2, p. 1056.

[19] I. Morrison and S. Nugrahanto, "Decision Support With BPEL and Web Services," *Int. J. Healthc. Inf. Syst. Informatics*, vol. 2, no. Hershey, PA, USA, pp. 67–74, 2007.

[20] Y. Huang, L. a Noirot, K. M. Heard, R. M. Reichley, W. C. Dunagan, and T. C. Bailey, "Migrating toward a next-generation clinical decision support application: the BJC HealthCare experience.," in *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2007, pp. 344–8.

[21] K. Kawamoto and D. F. Lobach, "Proposal for Fulfilling Strategic Objectives of the U.S. Roadmap for National Action on Decision Support through a Service-oriented Architecture Leveraging HL7 Services," *J. Am. Med. Informatics Assoc.*, vol. 14, no. 2, pp. 146–155, Mar. 2007.

[22] D. F. . F. Pires, C. A. C. . A. C. Teixeira, E. E. S. . E. S. d E. S. d Ruiz, R. Preto, R. W. Luis, S. Carlos, and W. Street, "A UMLS interoperable solution to support collaborative diagnosis decision making over the internet," in *Proceedings of the ACM Symposium on Applied Computing*, 2008, pp. 1400–1404.

[23] J. H. Jahnke-Weber, M. Price, and G. McCallum, "Making available Clinical Decision Support in Service-Oriented Architectures," *J. Inf. Technol. Healthc.*, vol. 6, no. 1, pp. 42–54, 2008.

[24] J. H. Jahnke-Weber, “Design of decoupled clinical decision support for service-oriented architectures,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 19, no. 2, pp. 159–183, Mar. 2009.

[25] A. Wright and D. F. Sittig, “SANDS: A service-oriented architecture for clinical decision support in a National Health Information Network,” *J. Biomed. Inform.*, vol. 41, no. 6, pp. 962–981, 2008.

[26] V. Kashyap, “From the Bench to the Bedside: The Role of Semantic Web and Translational Medicine for Enabling the Next Generation Healthcare Enterprise,” in *Biomedical Engineering Systems and Technologies SE - 3*, vol. 25, A. Fred, J. Filipe, and H. Gamboa, Eds. Springer Berlin Heidelberg, 2009, pp. 35–56.

[27] V. J. Zhu, S. J. Grannis, M. B. Rosenman, and S. M. Downs, “Implementing broad scale childhood immunization decision support as a web service.,” in *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2009, vol. 2009, pp. 745–9.

[28] J. Fdez-olivares, A. C. Juan, and L. Castillo, “OncoTheraper : Clinical Decision Support for Oncology Therapy Planning Based on Temporal Hierarchical Tasks Networks,” in *Knowledge Management for Health Care Procedures*, 2009, pp. 25–41.

[29] A. Riposan, I. J. Taylor, O. Rana, D. R. Owens, and E. C. Conley, “The TRIACS analytical workflows platform for distributed clinical decision support,” in *2009 22nd IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, 2009, p. 8 pp. –.

[30] G. B. Laleci and A. Dogac, “A Semantically Enriched Clinical Guideline Model Enabling Deployment in Heterogeneous Healthcare Environments,” *Inf. Technol. Biomed. IEEE Trans.*, vol. 13, no. 2, pp. 263–273, 2009.

[31] A. Kameas and I. Calemis, “Pervasive Systems in Health Care,” in *Handbook of Ambient Intelligence and Smart Environments SE - 12*, H. Nakashima, H. Aghajan, and J. Augusto, Eds. Springer US, 2010, pp. 315–346.

[32] A. Hatsek, Y. Shahar, M. Taieb-Maimon, E. Shalom, D. Klimov, and E. Lunenfeld, “A scalable architecture for incremental specification and maintenance of procedural and declarative clinical decision-support knowledge,” *Open Med. Inform. J.*, vol. 4, pp. 255–277, 2010.

[33] D. D. . Acosta, V. V. . Patkar, M. M. . Keshtgar, and J. . J. Fox, “Challenges in delivering decision support systems: The MATE experience,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5943 LNAI, pp. 124–140, 2010.

[34] G. Del Fiol, K. Kawamoto, N. M. A. Lapointe, E. L. Eisenstein, K. J. Anstrom, L. L. Wood, and D. F. Lobach, “Improving Medication Adherence in a Regional Healthcare Information Exchange using a Scalable, Claims-Driven, and Service-

Oriented Approach," in *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2010, vol. 2010, pp. 137–141.

[35] F. D. Backere, K. Steurbaut, F. D. Turck, K. Colpaert, and J. Decruyenaere, "On the Design of a Management Platform for Antibiotic Guidelines in the Intensive Care Unit," in *Proceedings Fifth International Conference on Software Engineering Advances (ICSEA 2010)*, 2010, pp. 406 – 11.

[36] K. Kawamoto, G. Del Fiol, C. Orton, and D. F. Lobach, "System-agnostic clinical decision support services: benefits and challenges for scalable decision support.," *Open Med. Inform. J.*, vol. 4, pp. 245–54, Jan. 2010.

[37] J. A. Kim, Shim, S. Kim, J. Lee, I. Cho, and Y. Kim, "Implementation of Guideline-Based CDSS," in *Proceedings of the 2011 International Conference on Ubiquitious Computing and Multimedia Applications (UCMA 2011)*, 2011, pp. 96 – 9.

[38] K. Fehre and K. P. Adlassnig, "Service-Oriented Arden-Sysntax-Based Clinical Decision Support," in *Ehealth2011: Health Informatics Meets Ehealth: Von Der Wissenschaft Zur Anwendung Und Zurueck: Grenzen Uberwinden. Continuity of Care*, 2011, pp. 123–128.

[39] V. Huser, L. V Rasmussen, R. Oberg, and J. B. Starren, "Implementation of workflow engine technology to deliver basic clinical decision support functionality.," *BMC Med. Res. Methodol.*, vol. 11, no. 1, p. 43, Jan. 2011.

[40] R. Kamaleswaran and C. McGregor, "CBPsp: Complex business processes for stream processing," in *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering: Vision for a Greener Future, CCECE 2012*, 2012.

[41] K. Kawamoto, J. Jacobs, B. M. Welch, V. Huser, M. D. Paterno, G. Del Fiol, D. Shields, H. R. Strasberg, P. J. Haug, Z. Liu, R. a Jenders, D. W. Rowed, D. Chertcoff, K. Fehre, K.-P. Adlassnig, and a C. Curtis, "Clinical information system services and capabilities desired for scalable, standards-based, service-oriented decision support: consensus assessment of the Health Level 7 clinical decision support Work Group.," in *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2012, vol. 2012, pp. 446–55.

[42] A. Mouttham, C. Kuziemsky, D. Langayan, L. Peyton, and J. Pereira, "Interoperable support for collaborative, mobile, and accessible health care," *Inf. Syst. Front.*, vol. 14, no. 1, pp. 73–85, 2012.

[43] R. Cucino and C. Eccher, "Modeling Healthcare Processes in BPEL: A Colon Cancer Screening Case Study," in *Electronic Healthcare SE - 10*, vol. 69, M. Szomszor and P. Kostkova, Eds. Springer Berlin Heidelberg, 2012, pp. 78–85.

[44] M. D. Paterno, H. S. Goldberg, L. Simonaitis, B. E. Dixon, A. Wright, B. H. Rocha, H. Z. Ramelson, and B. Middleton, "Using a service oriented architecture approach to clinical decision support: performance results from two CDS

Consortium demonstrations.," in *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2012, vol. 2012, pp. 690–698.

[45] M. Hussain, a. M. Khattak, W. a. Khan, I. Fatima, M. B. Amin, Z. Pervez, R. Batool, M. a. Saleem, M. Afzal, M. Faheem, M. H. Saddiqi, S. Y. Lee, and K. Latif, "Cloud-based Smart CDSS for chronic diseases," *Health Technol. (Berl).*, vol. 3, no. 2, pp. 153–175, Mar. 2013.

[46] M. Papazoglou, "What's in a Service?," in *Software Architecture SE - 3*, vol. 4758, F. Oquendo, Ed. Springer Berlin Heidelberg, 2007, pp. 11–28.

[47] K. Sartipi, M. H. Yarmand, and D. G. Down, "Mined-knowledge and decision support services in electronic health," in *Proceedings - ICSE 2007 Workshops: International Workshop on Systems Development in SOA Environments, SDSOA'07*, 2007.

[48] S. Colantonio, M. Martinelli, D. Moroni, O. Salvetti, A. Candelieri, D. Conforti, F. Perticone, A. Sciacqua, C. Biniaris, and A. Marsh, "A decision support resource as a kernel of a semantic Web based platform oriented to heart failure," in *Proceedings of the 2008 International Conference on Semantic Web and Web Services, SWWS 2008*, 2008, pp. 142–148.

[49] B. Van Den Bossche, S. Van Hoecke, C. Danneels, J. Decruyenaere, B. Dhoedt, and F. De Turck, "Design of a {JAIN} SLEE/ESB-based platform for routing medical data in the {ICU}," *Comput. Methods Programs Biomed.*, vol. 91, no. 3, pp. 265–277, 2008.

[50] K. Steurbaut, K. Colpaert, B. Gadeyne, P. Depuydt, P. Vosters, C. Danneels, D. Benoit, J. Decruyenaere, and F. De Turck, "COSARA: Integrated Service Platform for Infection Surveillance and Antibiotic Management in the ICU," *J. Med. Syst.*, vol. 36, no. 6, pp. 3765–3775, 2012.

[51] A. Benharref and M. Serhani, "Novel Cloud and SOA Based Framework for E-health Monitoring Using Wireless Biosensors," *Biomed. Heal. Informatics, IEEE J.*, vol. PP, no. 99, p. 1, 2013.

[52] A. Martínez-García, A. Moreno-Conde, F. Jódar-Sánchez, S. Leal, and C. Parra, "Sharing clinical decisions for multimorbidity case management using social network and open-source tools," *J. Biomed. Inform.*, no. 0, p. -, 2013.

[53] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe, and M. Stefanelli, "Comparing Computer-interpretable Guideline Models: A Case-study Approach," *J. Am. Med. Informatics Assoc.*, vol. 10, no. 1, pp. 52–68, Jan. 2003.

[54] J. Boyer and H. Mili, *Agile Business Rule Development. Process, Architecture, and JRules Examples*. Springer Berlin Heidelberg, 2011, pp. 251–257.

[55] P. Besana, V. Patkar, D. Glasspool, and D. Robertson, “Distributed Workflows: The OpenKnowledge Experience,” in *On the Move to Meaningful Internet Systems: OTM 2008 Workshops SE - 123*, vol. 5333, R. Meersman, Z. Tari, and P. Herrero, Eds. Springer Berlin Heidelberg, 2008, pp. 965–975.

[56] P. Besana, V. Patkar, A. Barker, D. Robertson, and D. Glasspool, “Sharing Choreographies in OpenKnowledge: A Novel Approach to Interoperability,” *J. Softw.*, vol. 4, no. 8, pp. 833–842, Oct. 2009.

[57] W3C, “Web Services Choreography Description Language, Version 1.0, W3C Candidate Recommendation, 9 November 2005.” [Online]. Available: http://www.w3.org/TR/ws-cdl-10/.

[58] T. Erl, *Service-Oriented Architecture Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

[59] Health Level 7, “HL7 Virtual Medical Record (vMR).” [Online]. Available: http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_(vMR).

[60] K. Kawamoto, A. Honey, and K. Rubin, “The HL7-OMG Healthcare Services Specification Project: Motivation, Methodology, and Deliverables for Enabling a Semantically Interoperable Service-oriented Architecture for Healthcare,” *J Am Med Inf. Assoc*, vol. 16, no. 6, pp. 874–881, 2009.

[61] P. A. de Clercq, J. A. Blom, H. H. M. Korsten, and A. Hasman, “Approaches for creating computer-interpretable guidelines that facilitate decision support,” *Artif. Intell. Med.*, vol. 31, no. 1, pp. 1–27, 2004.

[62] D. Isern and A. Moreno, “Computer-based execution of clinical guidelines: A review,” *Int. J. Med. Inform.*, vol. 77, no. 12, pp. 787–808, 2008.

[63] OASIS, “OASIS WS-BPEL Extension for People (BPEL4People) TC.” [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=bpel4people.

[64] F. Pontes Guimaraes, E. H. Kuroda, and D. M. Batista, “Performance Evaluation of Choreographies and Orchestrations with a New Simulator for Service Compositions,” *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2012 IEEE 17th International Workshop on*. pp. 140–144, 2012.

[65] OASIS, “Service Component Architecture Assembly Model Specification, Version 1.1, Committee Specification Draft 08 /Public Review Draft 03 31 May 2011.” [Online]. Available: http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-spec-v1.1-csprd03.pdf.

[66] M. Hinchey and L. Coyle, *Conquering Complexity*. London: Springer, 2012.

[67] The Apache Software Foundation, “Apache Tuscany.” [Online]. Available: http://tuscany.apache.org/.

[68] Fabric3, "Fabric3." [Online]. Available: http://www.fabric3.org/.

[69] OW2 Consortium, "FraSCAti." [Online]. Available: https://wiki.ow2.org/frascati/Wiki.jsp?page=FraSCAti.

[70] Red Hat Inc., "SwitchYard." [Online]. Available: http://www.jboss.org/switchyard.

[71] OMG, "Business process model and notation (BPMN 2.0)." [Online]. Available: http://www.omg.org/spec/BPMN/2.0/.

[72] M. Chinosi and A. Trombetta, "BPMN: An introduction to the standard," *Comput. Stand. Interfaces*, vol. 34, no. 1, pp. 124–134, Jan. 2012.

[73] M. P. Papazoglou and W.-J. Heuvel, "Service oriented architectures: approaches, technologies and research issues," *VLDB J.*, vol. 16, no. 3, pp. 389–415, Mar. 2007.

[74] F. Z. Safy, M. El-Ramly, and A. Salah, "Runtime Monitoring of SOA Applications: Importance, Implementations and Challenges," *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. pp. 315–319, 2013.

[75] U.S. Department of Health and Human Services, "What is health IT?" [Online]. Available: http://www.hrsa.gov/healthit/toolbox/oralhealthittoolbox/introduction/whatishealthit.html.

[76] B. Chaudhry, J. Wang, S. Wu, M. Maglione, W. Mojica, E. Roth, S. C. Morton, and P. G. Shekelle, "Systematic Review : Impact of Health Information Technology on Quality , Efficiency , and Costs of Medical Care," *Ann. Intern. Med.*, no. January, 2006.

[77] G. F. Anderson, B. K. Frogner, R. a Johns, and U. E. Reinhardt, "Health care spending and use of information technology in OECD countries.," *Health Aff. (Millwood).*, vol. 25, no. 3, pp. 819–31, 2006.

[78] D. Blumenthal, "Stimulating the Adoption of Health Information Technology," *N. Engl. J. Med.*, vol. 360, no. 15, pp. 1477–1479, Apr. 2009.

[79] The Department of Health, "Delivering 21 st Century IT Support for the NHS National Strategic Programme," London, 2002.

[80] D. A. Ludwick and J. Doucette, "Adopting electronic medical records in primary care: Lessons learned from health information systems implementation experience in seven countries," *Int. J. Med. Inform.*, vol. 78, no. 1, pp. 22–31, Jan. 2009.

[81] The Office of the National Coordinator for Health Information Technology, "Update on the adoption of health information technology and related efforts to

facilitate the electronic use and exchange of health information," Washington, D.C, 2013.

[82] K. M. Unertl, K. B. Johnson, and N. M. Lorenzi, "Health information exchange technology on the front lines of healthcare: workflow factors and patterns of use," *J. Am. Med. Informatics Assoc.*, vol. 19, no. 3, pp. 392–400, 2012.

[83] H. W, "Health care information technology: Progress and barriers," *JAMA*, vol. 292, no. 18, pp. 2273–2274, 2004.

[84] S. Hoffman and A. Podgurski, "Finding a cure: The case for regulation and oversight of electronic health record systems," *Hardvard J. Law Technol.*, vol. 22, no. 1, 2008.

[85] A.-M. Audet, M. M. Doty, J. Peugh, J. Shamasdin, K. Zapert, and S. Schoenbaum, "Information technologies: when will they make it into physicians' black bags?," *Medscape Gen. Med.*, vol. 6, no. 4, 2004.

[86] J. R. Vest and L. D. Gamm, "Health information exchange: persistent challenges and new strategies," *J. Am. Med. Informatics Assoc.*, vol. 17, no. 3, pp. 288–294, 2010.

[87] T. G. Kannampallil, G. F. Schauer, T. Cohen, and V. L. Patel, "Considering complexity in healthcare systems," *J. Biomed. Inform.*, vol. 44, no. 6, pp. 943–947, 2011.

[88] B.-T. Karsh, M. B. Weinger, P. A. Abbott, and R. L. Wears, "Health information technology: fallacies and sober realities," *J. Am. Med. Informatics Assoc.*, vol. 17, no. 6, pp. 617–623, 2010.

[89] U.S. Department of Health and Human Services, "HIPAA Privacy Rule, 45 CFR Parts 160 and 164, RIN 0945-AA03," 2013.

[90] P. Gibbons, N. Arzt, S. Burke-Beebe, C. Chute, G. Dickinson, T. Flewelling, T. Jepsen, D. Kamens, J. Larson, J. Ritter, M. Rozen, S. Selover, and J. Stanford, "Coming to Terms: Scoping Interoperability for Health Care," Health Level Seven International, 2007.

[91] Health Level Seven International (HL7), "HL7 Messaging Standard Version 2.7." [Online]. Available: http://www.hl7.org/.

[92] ASTM, "Committee E31 on Healthcare Informatics." [Online]. Available: http://www.astm.org/COMMITTEE/E31.htm.

[93] O. S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM)*. Springer, 2011.

[94] NEMA, "DICOM." [Online]. Available: http://medical.nema.org/.

[95] CDIS, "CDIS." [Online]. Available: http://www.cdisc.org/.

[96] N. C. US Institute, “Terminology Resources: NCI Enterprise Vocabulary Services (EVS), Dictionaries, FedMed, FDA, CDISC, and NCPDP Terminology.” [Online]. Available: http://www.cancer.gov/cancertopics/cancerlibrary/terminologyresources.

[97] OMG, “Object Magement Group.” [Online]. Available: http://www.omg.org/.

[98] Organization for the Advancement of Structured Information Standards, “OASIS.” [Online]. Available: https://www.oasis-open.org/.

[99] OASIS, “Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of Security Assertion Markup Language (SAML) for Healthcare Version 1.0.” [Online]. Available: http://docs.oasis-open.org/security/xspa/v1.0/saml-xspa-1.0-os.html.

[100] OASIS, “Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of WS-Trust for Healthcare Version 1.0.” [Online]. Available: http://docs.oasis-open.org/xspa/ws-trust-v1.0/xspa-ws-trust-profile-os.html.

[101] IHTSDO, “International Health Terminology Standards Development Organization.” [Online]. Available: http://www.ihtsdo.org/.

[102] T. Benson, *Principles of Health Interoperability Hl7 and Snomed*. Springer, 2009, p. 190.

[103] CEN, “European Committee for Standardization.” [Online]. Available: http://www.cen.eu.

[104] CEN, “CEN TC 251.” [Online]. Available: http://www.cen.eu/CEN/Sectors/TechnicalCommitteesWorkshops/CENTechnicalCommittees/Pages/Standards.aspx?param=6232&title=CEN/TC�251.

[105] ISO, “TC 215.” [Online]. Available: http://www.iso.org/iso/iso_technical_committee?commid=54960.

[106] NCHS, “National Center for Health Statistics.” [Online]. Available: http://www.cdc.gov/nchs/.

[107] B. Sharma, “Electronic Healthcare Maturity Model (eHMM): A White paper,” Chennai, India, 2008.

[108] D. Garets and M. Davis, “Electronic Medical Records vs . Electronic Health Records : Yes , There Is a Difference.,” *Policy white Pap. Chicago, HIMSS Anal.*, 2006.

[109] HIMSS Analytics, “Electronic Medical Record Adoption Model (EMRAM).” [Online]. Available: http://www.himssanalytics.org/emram/emram.aspx.

[110] D. Wainwright and T. Waring, “The information management and technology strategy of the UK National Health Service – Determining progress in the NHS

acute hospital sector," *Int. J. Public Sect. Manag.*, vol. 13, no. 3, pp. 241–259, 2000.

[111] D. Blumenthal and M. Tavenner, "The 'Meaningful Use' Regulation for Electronic Health Records," *N. Engl. J. Med.*, vol. 363, no. 6, pp. 501–504, 2010.

[112] R. Graham, M. Mancher, D. M. Wolman, S. Greenfield, and E. Steinberg, *Clinical Practice Guidelines We Can Trust*. National Academies Press, 2011.

[113] M. Musen, B. Middleton, and R. Greenes, "Clinical Decision-Support Systems," in *Biomedical Informatics*, E. H. Shortliffe and J. J. Cimino, Eds. Springer London, 2014, pp. 643–674.

[114] J. Cimino and G. Del Fiol, "Infobuttons and point of care access to knowledge," in *Clinical decision support-the road ahead*, 2007, pp. 345–372.

[115] E. Berner and T. La Lande, "Overview of Clinical Decision Support Systems," in *Clinical Decision Support Systems*, E. Berner, Ed. Springer New York, 2007, pp. 3–22.

[116] A. Wright and D. F. Sittig, "A four-phase model of the evolution of clinical decision support architectures," *Int. J. Med. Inform.*, vol. 77, no. 10, pp. 641–649, 2008.

[117] T. J. Bright, A. Wong, R. Dhurjati, E. Bristow, L. Bastian, R. R. Coeytaux, G. Samsa, V. Hasselblad, J. W. Williams, M. D. Musty, L. Wing, A. S. Kendrick, G. D. Sanders, and D. Lobach, "Effect of Clinical Decision-Support Systems A Systematic Review," *Ann. Intern. Med.*, vol. 157, no. 1, pp. 29–43, Jul. 2012.

[118] H. Kashfi, "The intersection of clinical decision support and electronic health record: A literature review," in *2011 Federated Conference on Computer Science and Information Systems, FedCSIS 2011*, 2011, pp. 347–353.

[119] J. Fox, D. Glasspool, V. Patkar, M. Austin, L. Black, M. South, D. Robertson, and C. Vincent, "Delivering clinical decision support services: there is nothing as practical as a good theory.," *J. Biomed. Inform.*, vol. 43, no. 5, pp. 831–43, Oct. 2010.

[120] C. Combi, M. Gozzi, J. M. Juarez, B. Oliboni, and G. Pozzi, "Conceptual Modeling of Temporal Clinical Workflows," *Temporal Representation and Reasoning, 14th International Symposium on*. pp. 70–81, 2007.

[121] S. SR, K. R, C. PD, and et al, "Physicians and electronic health records: A statewide survey," *Arch. Intern. Med.*, vol. 167, no. 5, pp. 507–512, 2007.

[122] D. F. Sittig, A. Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, E. Campbell, and D. W. Bates, "Grand challenges in clinical decision support," *J. Biomed. Inform.*, vol. 41, no. 2, pp. 387–392, 2008.

[123] W. M. P. Aalst and A. H. M. Hofstede, “Workflow patterns put into context,” *Softw. Syst. Model.*, vol. 11, no. 3, pp. 319–323, 2012.

[124] M. Samwald, K. Fehre, J. de Bruin, and K.-P. Adlassnig, “The Arden Syntax standard for clinical decision support: Experiences and directions,” *J. Biomed. Inform.*, no. 0, p. -, 2012.

[125] P. Nadkarni, “Challenges in Electronic Decision Support Implementation,” in *Metadata-driven Software Systems in Biomedicine*, Springer London, 2011, pp. 123–142.

[126] HL7, “GELLO (HL7 Version 3 Standard: Gello: A Common Expression Language, Release 2).” [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=5.

[127] K. Kawamoto, G. Del Fiol, H. R. Strasberg, N. Hulse, C. Curtis, J. J. Cimino, B. H. Rocha, S. Maviglia, E. Fry, H. J. Scherpbier, V. Huser, P. K. Redington, D. K. Vawdrey, J.-C. Dufour, M. Price, J. H. Weber, T. White, K. S. Hughes, J. C. McClay, C. Wood, K. Eckert, S. Bolte, D. Shields, P. R. Tattam, P. Scott, Z. Liu, and A. K. McIntyre, “Multi-National, Multi-Institutional Analysis of Clinical Decision Support Data Needs to Inform Development of the HL7 Virtual Medical Record Standard.,” *AMIA Annu. Symp. Proc.*, vol. 2010, pp. 377–81, Jan. 2010.

[128] HL7, “Health Level Seven International.” [Online]. Available: http://www.hl7.org/.

[129] Health Level Seven International (HL7), “HL7 Decision Support Service (DSS).” [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12.

[130] K. Kawamoto, “OpenCDS.” [Online]. Available: http://www.opencds.org/.

[131] Health Level Seven International (HL7), “HL7 Version 3 Standard: Retrieve, Locate, and Update Service (RLUS) Release 1.” [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=89.

[132] International Organization for Standardization (ISO), “ISO/IEC/IEEE 42010:2007, Systems and software engineering -- Recommended practice for architectural description of software-intensive systems,” 2007. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=45991.

[133] The Open Group, “TOGAF 9.1, Introduction, Core Concepts.” [Online]. Available: http://pubs.opengroup.org/architecture/togaf91-doc/arch/chap02.html.

[134] MIT CISR, “MIT Center for Information System Research, Enterprise Architecture definition.” [Online]. Available: http://cisr.mit.edu/research/research-overview/classic-topics/enterprise-architecture/.

[135] B. H. Cameron and E. Mcmillan, “Analyzing the Current Trends in Enterprise Architecture Frameworks,” *J. Enterp. Archit.*, no. February, 2013.

[136] The Open Group, “TOGAF - Frequently Asked Questions.” [Online]. Available: http://www.opengroup.org/public/arch/p1/togaf_faq.htm.

[137] J. A. Zachman, “Enterprise Architecture : The Issue of the Century,” *Database Program. Des.*, pp. 44–53, 1996.

[138] The Open Group, “The Open Group Architecture Framework (TOGAF).” [Online]. Available: http://www.opengroup.org/togaf/.

[139] Executive Office of the President of the United States, “The Common Approach to Federal Enterprise Architecture.” [Online]. Available: http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/common_approach_to_federal_ea.pdf.

[140] Chief Information Officer U.S. Department of Defense, “DoD Architecture Framework (DoDAF).” [Online]. Available: http://dodcio.defense.gov/dodaf20.aspx.

[141] Gartner Group, “Gartner’s Enterprise Architecture Process and Framework Help Meet 21st Century Challenges.” [Online]. Available: https://www.gartner.com/doc/486650/gartners-enterprise-architecture-process-framework.

[142] The Open Group, “ArchiMate®.” [Online]. Available: http://www.opengroup.org/subjectareas/enterprise/archimate.

[143] M. M. Lankhorst, H. A. Proper, and H. Jonkers, “The Architecture of the ArchiMate Language,” in *Enterprise, Business-Process and Information Systems Modeling SE - 30*, vol. 29, T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, and R. Ukor, Eds. Springer Berlin Heidelberg, 2009, pp. 367–380.

[144] Advancing Open Standards for the Information Society (OASIS), “Reference Model for Service Oriented Architecture 1.0,” 2006. [Online]. Available: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.

[145] M. P. Papazoglou, “Service-oriented computing: concepts, characteristics and directions,” *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. pp. 3–12, 2003.

[146] P. Malinverno, D. C. Plummer, and G. Van Huizen, “Magic Quadrant for Application Services Governance,” 2013.

[147] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley, “SOMA: A method for developing service-oriented solutions,” *IBM Syst. J.*, vol. 47, no. 3, pp. 377–396, 2008.

[148] T. Erl, *SOA Design Patterns*. 2009.

[149] W. M. P. van der Aalst, “Business Process Management: A Comprehensive Survey,” *ISRN Softw. Eng.*, vol. 2013, pp. 1–37, 2013.

[150] S. Brahe, “BPM on Top of SOA: Experiences from the Financial Industry,” *Springer*, vol. 4714/2007, 2007.

[151] M. Dumas, M. La Rosa, J. Mendling, and H. Reijers, “Introduction to Business Process Management,” in *Fundamentals of Business Process Management*, Springer Berlin Heidelberg, 2013, pp. 1–31.

[152] M. Muehlen and J. Recker, “How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation,” in *Advanced Information Systems Engineering*, vol. 5074, Z. Bellahs�ne and M. L�onard, Eds. Springer Berlin Heidelberg, 2008, pp. 465–479.

[153] J. Boyer and H. Mili, “Introduction to Business Rules,” in *Agile Business Rule Development*, Springer Berlin Heidelberg, 2011, pp. 3–25.

[154] Businees Rules Group, “Defining Business Rules ~ What Are They Really?” [Online]. Available: http://www.businessrulesgroup.org/first_paper/br01c0.htm.

[155] M. zur Muehlen and M. Indulska, “Modeling languages for business processes and business rules: A representational analysis,” *Inf. Syst.*, vol. 35, no. 4, pp. 379–390, Jun. 2010.

[156] J. Boyer and H. Mili, “Issues in Rule Authoring,” in *Agile Business Rule Development*, Springer Berlin Heidelberg, 2011, pp. 245–281.

[157] R. Singer and E. Zinser, “Business Process Management — S-BPM a New Paradigm for Competitive Advantage?,” in *S-BPM ONE – Setting the Stage for Subject-Oriented Business Process Management SE - 5*, vol. 85, H. Buchwald, A. Fleischmann, D. Seese, and C. Stary, Eds. Springer Berlin Heidelberg, 2010, pp. 48–70.

[158] M. Weske, *Business Process Management*. Springer-Verlag Berlin Heidelberg, 2012.

[159] D. Schall, “Service Oriented Protocols for Human Computation,” in *Handbook of Human Computation SE - 42*, P. Michelucci, Ed. Springer New York, 2013, pp. 551–559.

[160] Advancing Open Standards for the Information Society (OASIS), “Web Services Business Process Execution Language Version 2 . 0,” 2007. .

[161] Advancing Open Standards for the Information Society (OASIS), “WS-BPEL Extension for People (BPEL4People) Specification Version 1.1,” 2010. [Online]. Available: http://docs.oasis-open.org/bpel4people/bpel4people-1.1-spec-cs-01.pdf.

[162] Advancing Open Standards for the Information Society (OASIS), “Web Services – Human Task (WS-HumanTask) Specification Version 1.1,” 2010. [Online]. Available: http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.pdf.

[163] L. Bonetta, “Whole-Genome Sequencing Breaks the Cost Barrier,” *Cell*, vol. 141, no. 6, pp. 917–919, 2010.

[164] I. S. Chan and G. S. Ginsburg, “Personalized medicine: progress and promise.,” *Annu. Rev. Genomics Hum. Genet.*, vol. 12, pp. 217–44, Sep. 2011.

[165] E. P. Bottinger, “Foundations, promises and uncertainties of personalized medicine,” *Mt. Sinai J. Med. A J. Transl. Pers. Med.*, vol. 74, no. 1, pp. 15–21, 2007.

[166] B. Welch and K. Kawamoto, “The Need for Clinical Decision Support Integrated with the Electronic Health Record for the Clinical Application of Whole Genome Sequencing Information,” *J. Pers. Med.*, vol. 3, no. 4, pp. 306–325, Dec. 2013.

[167] B. M. Welch, S. R. Loya, K. Eilbeck, and K. Kawamoto, “A Proposed Clinical Decision Support Architecture Capable of Supporting Whole Genome Sequence Information,” *J. Pers. Med.*, vol. 4, no. 2, pp. 176–199, 2014.

[168] B. M. Welch, K. Eilbeck, G. Del Fiol, L. Meyer, and K. Kawamoto, “Technical desiderata for the integration of genomic data with clinical decision support [Under review],” 2014.

[169] D. R. Masys, G. P. Jarvik, N. F. Abernethy, N. R. Anderson, G. J. Papanicolaou, D. N. Paltoo, M. A. Hoffman, I. S. Kohane, and H. P. Levy, “Technical desiderata for the integration of genomic data into Electronic Health Records,” *J. Biomed. Inform.*, vol. 45, no. 3, pp. 419–422, 2012.

[170] “Lynch syndrome.” [Online]. Available: http://ghr.nlm.nih.gov/condition/lynch-syndrome.

[171] “Lynch Syndrome Management.” [Online]. Available: http://genefacts.org/index.php?option=com_content&view=article&id=492:management&catid=113:lynch-syndrome&Itemid=665.

[172] National Center for Biotechnology Information, “Single Nucleotide Polymorphism Database.” [Online]. Available: http://www.ncbi.nlm.nih.gov/SNP/.

[173] O. Horaitis and R. G. H. Cotton, “The challenge of documenting mutation across the genome: The human genome variation society approach,” *Hum. Mutat.*, vol. 23, no. 5, pp. 447–452, May 2004.

[174] National Center for Biotechnology Information, “ClinVar.” [Online]. Available: http://www.ncbi.nlm.nih.gov/clinvar/.

[175] G. Wierda, *Mastering Archimate, Edition II*. R&A, The Netherlands, 2014.

[176] W. M. Van der Aalst, M. Dumas, and A. H. ter Hofstede, “Web service composition languages: old wine in new bottles?,” in *Euromicro Conference, 2003. Proceedings. 29th*, 2003, pp. 298–305.

[177] D. W. Bates, G. J. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton, “Ten Commandments for Effective Clinical Decision Support: Making the Practice of Evidence-based Medicine a Reality,” *J. Am. Med. Informatics Assoc.*, vol. 10, no. 6, pp. 523–530, Nov. 2003.

[178] National Institute for Health and Care Excellence, “NICE.” [Online]. Available: http://www.nice.org.uk/.

[179] WHO, “The Clinical Guideline Programme of the National Institute for Health and Clinical Excellence (NICE), A review bt the World Health Organization,” 2006.

[180] NCGC, “National Clinical Guideline Centre.” [Online]. Available: http://www.ncgc.ac.uk/.

[181] NCCC, “The National Collaborating Centre for Cancer.” [Online]. Available: http://www.wales.nhs.uk/sites3/home.cfm?orgid=432.

[182] NCC-WCH, “National Collaborating Centre for Women’s and Children's Health.” [Online]. Available: http://www.ncc-wch.org.uk/.

[183] NCCMH, “National Collaborating Centre for Mental Health.” [Online]. Available: http://www.nccmh.org.uk/.

[184] National Institute for Health and Care Excellence, “Reviewing published clinical guidelines.” [Online]. Available: http://www.nice.org.uk/aboutnice/whatwedo/aboutclinicalguidelines/ReviewingPublishedCGs.jsp.

[185] D. L. Sackett, W. M. C. Rosenberg, J. A. M. Gray, R. B. Haynes, and W. S. Richardson, “Evidence based medicine: what it is and what it isn’t,” *BMJ*, vol. 312, no. 7023, pp. 71–72, 1996.

[186] A. L. Francke, M. C. Smit, A. J. E. de Veer, and P. Mistiaen, “Factors influencing the implementation of clinical guidelines for health care professionals: a systematic meta-review.,” *BMC Med. Inform. Decis. Mak.*, vol. 8, no. 1, p. 38, Jan. 2008.

[187] A. Latoszek-Berendsen, H. Tange, H. J. van den Herik, and A. Hasman, “From Clinical Practice Guidelines to Computer-interpretable Guidelines. A Literature Overview,” *Methods Inf. Med.*, vol. 49, no. 6, pp. 550–570, 2010.

[188] J. Boyer and H. Mili, *Agile Business Rule Development, Process, Architecture, and JRules Examples*. Springer Berlin Heidelberg, 2011.

[189] National Institute for Health and Care Excellence, “NICE Pathways, COPD Overview.” [Online]. Available: http://guidance.nice.org.uk/CG101.

[190] “Oxygen XML Editor.” [Online]. Available: http://www.oxygenxml.com/.

[191] National Institute for Health and Care Excellence, “Lung cancer (CG121).” [Online]. Available: http://www.nice.org.uk/CG121.

[192] National Institute for Health and Care Excellence, “NICE Pathways, Lung Cancer Overview.” [Online]. Available: http://pathways.nice.org.uk/pathways/lung-cancer.

[193] E. Hollnagel and D. D. Woods, *Joint cognitive systems: Foundations of cognitive systems engineering*. CRC Press, 2005.

[194] A. Hoffer, “BPM: antidote to inefficiency. Business process management serves as a foundation for effective workflow automation in healthcare,” *Health Manag. Technol.*, vol. 24, no. 1, p. 40—43, Jan. 2003.

[195] “Evolution of Enterprise Systems Architectures,” in *Business Process Management SE - 2*, Springer Berlin Heidelberg, 2007, pp. 25–69.

[196] R. Lenz and M. Reichert, “IT support for healthcare processes – premises, challenges, perspectives,” *Data Knowl. Eng.*, vol. 61, no. 1, pp. 39–58, Apr. 2007.

[197] JBoss Community, “jBPM.” [Online]. Available: http://www.jboss.org/jbpm.

[198] S. M. Schneider, A. F. Gardner, L. D. Weiss, J. P. Wood, M. Ybarra, D. M. Beck, A. R. Stauffer, D. Wilkerson, T. Brabson, A. Jennings, M. Mitchell, R. B. McGrath, T. A. Christopher, B. King, R. L. Muelleman, M. J. Wagner, D. M. Char, D. L. McGee, R. L. Pilgrim, J. B. Moskovitz, A. R. Zinkel, M. Byers, W. T. Briggs, C. D. Hobgood, D. F. Kupas, J. Kruger, C. J. Stratford, and N. Jouriles, “The Future of Emergency Medicine,” *Acad. Emerg. Med.*, vol. 17, no. 9, pp. 998–1003, 2010.

[199] D. B. Chalfin, S. Trzeciak, A. Likourezos, B. M. Baumann, and R. P. Dellinger, “Impact of delayed transfer of critically ill patients from the emergency department to the intensive care unit.,” *Crit. Care Med.*, vol. 35, no. 6, pp. 1477–83, Jun. 2007.

[200] J. M. Pines, J. A. Hilton, E. J. Weber, A. J. Alkemade, H. Al Shabanah, P. D. Anderson, M. Bernhard, A. Bertini, A. Gries, S. Ferrandiz, V. A. Kumar, V.-P. Harjola, B. Hogan, B. Madsen, S. Mason, G. Öhlén, T. Rainer, N. Rathlev, E. Revue, D. Richardson, M. Sattarian, and M. J. Schull, “International Perspectives on Emergency Department Crowding,” *Acad. Emerg. Med.*, vol. 18, no. 12, pp. 1358–1370, 2011.

[201] L. I. Horwitz, J. Green, and E. H. Bradley, "US Emergency Department Performance on Wait Time and Length of Visit," *Ann. Emerg. Med.*, vol. 55, no. 2, pp. 133–141, Feb. 2010.

[202] C. M. Fernandes, J. M. Christenson, and a Price, "Continuous quality improvement reduces length of stay for fast-track patients in an emergency department.," *Acad. Emerg. Med.*, vol. 3, no. 3, pp. 258–63, Mar. 1996.

[203] J. L. Wiler, C. Gentle, J. M. Halfpenny, A. Heins, A. Mehrotra, M. G. Mikhail, and D. Fite, "Optimizing Emergency Department Front-End Operations," *Ann. Emerg. Med.*, vol. 55, no. 2, pp. 142–160.e1, Feb. 2010.

[204] National Institute for Health and Care Excellence, "Acutely ill patients in hospital," 2007. [Online]. Available: http://www.nice.org.uk/guidance/cg50/resources/cg50-acutely-ill-patients-in-hospital-full-guideline3.

[205] F. Marir and J. Ndeta, "Workflow Design Patterns for Developing and Maintaing e-Business Workflow Systems," in *BPM and Workflow Handbook: Spotlight on Business Intelligence*, L. Fischer, Ed. FL: Future Strategies Inc., Book Division, 2010, pp. 231–244.

[206] Advancing Open Standards for the Information Society (OASIS), "OASIS SOA Repository Artifact Model and Protocol (S-RAMP) TC." [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=s-ramp#announcements.

[207] B. Kitchenham, "Procedures for Performing Systematic Reviews. Technical Report TR/SE-0401 and NICTA Tecnical Report 0400011T1." Keele University and National ICT Australia Ltd, 2004.

[208] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *BMJ*, vol. 339, 2009.

[209] T. Dyba, T. Dingsoyr, and G. K. Hanssen, "Applying Systematic Reviews to Diverse Study Types: An Experience Report," *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. pp. 225–234, 2007.

[210] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Durham, 2007.

[211] J. F. Burnham, "Scopus database: a review.," *Biomed. Digit. Libr.*, vol. 3, p. 1, Jan. 2006.

Appendix A: Systematic Review Methods

1.1. Methods

The systematic review followed the process suggested by Kitchenham [207] and in accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [208]. The process followed three steps: a) planning, b) conduction, and c) reporting. In the planning step, the study questions and review protocol were established. In the conduction step, the studies were identified, selected, and evaluated according to the inclusion and exclusion criteria defined during the planning phase. Finally, during the reporting step the final report was created and presented.

Data Sources and Searches

The following electronic bibliographic databases were included during this systematic review:

- ACM Digital Library
- Compendex
- IEEE Explore
- Science Direct
- Springer
- Web of Science
- Scopus

The electronic search covered all English literature until 20 October 2013. The first six databases are considered efficient to conduct systematic reviews in the context of software engineering [209]. The database Scopus was included because it is considered the largest database of abstracts and citations [210]. Scopus encompasses approximately 4600 health science titles, includes 100% MEDLINE and EMBASE coverage [211], and provides coverage of medical journals.

The main keywords used during the electronic search were “Service Oriented” and “Clinical Decision Support.” The following related terms were included:

- **service oriented**: "service-oriented", "service based", "service-based", "service orientation", "service-orientation", "SOC" and "SOA"
- **clinical decision support**: "CDS" and "CDSS"

The following search string was applied and adapted to each publications database:

("*service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation" OR "SOC" OR "SOA") AND ("clinical decision support" AND "CDS" AND "CDSS*")

Search strings from each database

ACM Digital Library

The search in the ACM Digital Library was performed on October 16, 2013 and four studies were obtained. Because of the options available in this database the search string had to be divided into the following two substrings:

Table C.1: Search strings used on the ACM Digital Library database.

Search string	**Number of publications obtained**
(Abstract:"service oriented" OR Abstract:"service-oriented" OR Abstract:"service based" OR Abstract:"service-based" OR Abstract:"service orientation" OR Abstract:"service-orientation" OR Abstract:"SOC" OR Abstract:"SOA") AND (Abstract:"Clinical Decision Support" OR Abstract:"CDS" OR Abstract:"CDSS")	3
(Title:"service oriented" OR Title:"service-oriented" OR Title:"service based" ORTitle:"service-based" OR Abstract:"service orientation" ORAbstract:"service-orientation" OR Title:"SOC" OR Title:"SOA") AND (Title:"Clinical Decision Support" OR Title:"CDS" ORTitle:"CDSS")	1

Compedex

The search in the Compendex database was performed on October 16, 2013 and 80 studies were returned. The search was conducted in the subject, title and abstract of all the databases available. The result was filtered by language, considering only studies written in English. The defined search is presented in the following table.

Table C.2: Search string used on Compendex.

Search string	Number of publications obtained
((("service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation" OR "SOA" OR "SOC") WN KY) AND (("clinical decision support" OR "CDS" OR "CDSS") WN KY)), English only	80

IEEE Explore

Since IEEE Xplore does now allow searching in the abstract and title in the same string, the search string was divided into the following two substrings. The search was performed on October 16, 2013 and 13 studies were returned.

Table C.3: Search string used on IEEE Xplore.

Search strings	Number of publications obtained
("Abstract":"service oriented" OR "Abstract":"service-oriented" OR "Abstract":"service based" OR "Abstract":"service-based" OR "Abstract":"service orientation" OR "Abstract":"service-orientation" OR "Abstract":"SOC" OR "Abstract":"SOA") AND ("Abstract":"Clinical decision support" OR "Abstract":"CDS" OR "Abstract":"CDSS")	13
("Document Title":"service oriented" OR "Document Title":"service-oriented" OR "Document Title":"service based" OR "Document Title":"service-based" OR "Document Title":"service orientation" OR "Document Title":"service-orientation" OR "Document Title":"SOC" OR "Document Title":"SOA") AND ("Document Title":"Clinical decision support" OR "Document Title":"CDS" OR "Document Title":"CDSS")	0

ScienceDirect

The search in ScienceDirect was performed on October 17, 2013. The title, abstract, and keywords were included and 12 studies were returned. The following table shows the search string and number of publications retrieved.

Table C.4: Search string used on ScienceDirect.

Search strings	**Number of publications obtained**
TITLE-ABSTR-KEY("service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation" OR "SOC" OR "SOA") and TITLE-ABSTR-KEY("clinical decision support" OR "CDS" OR "CDSS")	12

Scopus

The search in Scopus was performed on October 17, 2013 and 79 studies were retrieved. The following table shows the string used in this database.

Table C.5: Search string used on Scopus.

Search strings	**Number of publications obtained**
TITLE-ABS-KEY(("service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation" OR "SOC" OR "SOA") AND ("clinical decision support" OR "CDS" OR "CDSS"))	79

Springer

The search in Springer was performed on October 17, 2013. The following table shows the string used in the Springer search engine.

Table C.6: Search string used on Springer.

Search strings	**Number of publications obtained**
("service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation") AND ("clinical decision support")	118

The terms "CDS" and "CDSS" were removed from the search string because several unrelated studies were obtained (393). This is because Springer does not provide a search tool that could limit the search to the abstract and title; it does search in the whole content of the studies.

Web of Science

The search in Web of Science was performed on October 17, 2013 and a set of 48 articles was obtained. Since this database does not allow searching by abstract, the search was performed using the title and "topic". The defined search string is presented in the following table.

Table C.7: Search string used on Web of Science.

Search strings	**Number of publications obtained**
Topic=(("service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation" OR "SOC" OR "SOA") AND ("clinical decision support" OR "CDS" OR "CDSS")) OR Title=(("service oriented" OR "service-oriented" OR "service based" OR "service-based" OR "service orientation" OR "service-orientation" OR "SOC" OR "SOA") AND ("clinical decision support" OR "CDS" OR "CDSS"))	48

Total of search results

The following table C.8 shows the total of articles obtained from the search.

Table C.8: Search results.

Database or search engine	**Number of search results**
ACM Digital Library	4
Compendex	80
IEEE Explore	13
ScienceDirect	12
Scopus	79
Springer	118
Web of Science	48
Combined studies	354
Repeated studies	216
Total of studies after removing repeated results	**138**

Study Selection

The review only includes studies written in English. The inclusion and exclusion criteria adopted during the review are described below:

Inclusion criteria (at least one required):

- The study proposes or reports on the design and development of service-oriented CDS.
- The study proposes or reports on a new technology for developing service-oriented CDS systems.
- The study proposes or reports on a process, method, technique, or reference architecture that supports either the design or the development of service-oriented CDS systems.
- The study proposes or reports on a healthcare standard that supports either the design or development of service-oriented CDS systems.

Exclusion criteria:

- The study proposes or reports on the design of CDS systems without using service orientation.
- The study presents contributions in areas other than CDS.
- The study is a table of contents, short course description, tutorial, copyright form or a conference or workshop agenda.

The study selection process consisted of two stages. In the first stage the researcher selected the studies based on the title, abstracts and keywords. In the second stage, each article was read in full and analysed. Additionally, relevant cited works were also included in the review.

Data Extraction and Quality Assessment

Information related to architectural approaches, technology used for the implementation, HIT standards, and challenges and potential solutions were extracted and categorized.

Data Synthesis and Analysis

The data was tabulated and summarized. The extracted information was categorized into the following groups:

- Publications per year.
- Architectural approach.
- Healthcare standards used for the implementation.
- Challenges and lessons learned.

List of studies included in the systematic review

1. Acosta, D. D. ., Patkar, V. V. ., Keshtgar, M. M. ., & Fox, J. . J. (2010). Challenges in delivering decision support systems: The MATE experience. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5943 LNAI, 124–140. doi:10.1007/978-3-642-11808-1_11
2. Backere, F. D., Steurbaut, K., Turck, F. D., Colpaert, K., & Decruyenaere, J. (2010). On the Design of a Management Platform for Antibiotic Guidelines in the Intensive Care Unit. In Proceedings Fifth International Conference on Software Engineering Advances (ICSEA 2010) (pp. 406 – 11). Los Alamitos, CA, USA. Retrieved from http://dx.doi.org/10.1109/ICSEA.2010.69
3. Benharref, A., & Serhani, M. (2013). Novel Cloud and SOA Based Framework for E-health Monitoring Using Wireless Biosensors. Biomedical and Health Informatics, IEEE Journal of, PP(99), 1. doi:10.1109/JBHI.2013.2262659
4. Besana, P., Patkar, V., Barker, A., Robertson, D., & Glasspool, D. (2009). Sharing Choreographies in OpenKnowledge: A Novel Approach to Interoperability. Journal of Software, 4(8), 833–842. doi:10.4304/jsw.4.8.833-842
5. Besana, P., Patkar, V., Glasspool, D., & Robertson, D. (2008). Distributed Workflows: The OpenKnowledge Experience. In R. Meersman, Z. Tari, & P. Herrero (Eds.), On the Move to Meaningful Internet Systems: OTM 2008 Workshops SE - 123 (Vol. 5333, pp. 965–975). Springer Berlin Heidelberg. doi:10.1007/978-3-540-88875-8_123
6. Bicer, V., Kilic, O., Dogac, A., & Laleci, G. B. (2005). Archetype-Based Semantic Interoperability of Web Service Messages in the Health Care Domain. Int'l Journal on Semantic Web & Information Systems, 1(December), 1–22.

7. Blackford MIDDLETON. (2009). The Clinical Decision Support Consortium *. In Studies in Health Technology and Informatics (pp. 26–30). doi:10.3233/978-1-60750-044-5-26
8. Bossche, B. Van Den, Hoecke, S. Van, Danneels, C., Decruyenaere, J., Dhoedt, B., & Turck, F. De. (2008). Design of a {JAIN} SLEE/ESB-based platform for routing medical data in the {ICU}. Computer Methods and Programs in Biomedicine, 91(3), 265–277. doi:http://dx.doi.org/10.1016/j.cmpb.2008.05.003
9. Catley, C., Petriu, D. C., & Frize, M. (2004). Software performance engineering of a Web service-based clinical decision support infrastructure. In Softw. Eng. Notes (USA) (Vol. 29, pp. 130 – 8). USA. Retrieved from http://dx.doi.org/10.1145/974043.974066
10. Ciccarese, P., Caffi, E., Boiocchi, L., Quaglini, S., & Stefanelli, M. (2004). A Guideline Management System. Medinfo, 28–32.
11. Colantonio, S., Martinelli, M., Moroni, D., Salvetti, O., Candelieri, A., Conforti, D., … Marsh, A. (2008). A decision support resource as a kernel of a semantic Web based platform oriented to heart failure. In Proceedings of the 2008 International Conference on Semantic Web and Web Services, SWWS 2008 (pp. 142–148). Las Vegas, NV, United states.
12. Cucino, R., & Eccher, C. (2012). Modeling Healthcare Processes in BPEL: A Colon Cancer Screening Case Study. In M. Szomszor & P. Kostkova (Eds.), Electronic Healthcare SE - 10 (Vol. 69, pp. 78–85). Springer Berlin Heidelberg. doi:10.1007/978-3-642-23635-8_10
13. Del Fiol, G., Kawamoto, K., Lapointe, N. M. A., Eisenstein, E. L., Anstrom, K. J., Wood, L. L., & Lobach, D. F. (2010). Improving Medication Adherence in a Regional Healthcare Information Exchange using a Scalable, Claims-Driven, and Service-Oriented Approach. In AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium (Vol. 2010, pp. 137–141). Retrieved from <Go to ISI>://MEDLINE:21346956
14. Fdez-olivares, J., Juan, A. C., & Castillo, L. (2009). OncoTheraper : Clinical Decision Support for Oncology Therapy Planning Based on Temporal Hierarchical Tasks Networks. In Knowledge Management for Health Care Procedures (pp. 25–41).
15. Fehre, K., & Adlassnig, K. P. (2011). Service-Oriented Arden-Sysntax-Based Clinical Decision Support. In G. Schreier, D. Hayn, & E. Ammenwerth (Eds.), Ehealth2011: Health Informatics Meets Ehealth: Von Der Wissenschaft Zur Anwendung Und

Zurueck: Grenzen Uberwinden. Continuity of Care (pp. 123–128). Retrieved from <Go to ISI>://WOS:000303910500018

16. Hatsek, A., Shahar, Y., Taieb-Maimon, M., Shalom, E., Klimov, D., & Lunenfeld, E. (2010). A scalable architecture for incremental specification and maintenance of procedural and declarative clinical decision-support knowledge. The open medical informatics journal, 4, 255–277. doi:10.2174/1874431101004010255
17. Heard, K. M., Huang, C., Noirot, L. a, Reichley, R. M., & Bailey, T. C. (2006). Using BPEL to define an executable CDS rule process. In AMIA Annual Symposium Proceedings (p. 947). American Medical Informatics Association. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1839607&tool=pmcentrez&rendertype=abstract
18. Huang, Y., Noirot, L. a, Heard, K. M., Reichley, R. M., Dunagan, W. C., & Bailey, T. C. (2007). Migrating toward a next-generation clinical decision support application: the BJC HealthCare experience. In AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium (pp. 344–8). Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2655891&tool=pmcentrez&rendertype=abstract
19. Huser, V., Rasmussen, L. V, Oberg, R., & Starren, J. B. (2011). Implementation of workflow engine technology to deliver basic clinical decision support functionality. BMC medical research methodology, 11(1), 43. doi:10.1186/1471-2288-11-43
20. Hussain, M., Khattak, a. M., Khan, W. a., Fatima, I., Amin, M. B., Pervez, Z., … Latif, K. (2013). Cloud-based Smart CDSS for chronic diseases. Health and Technology, 3(2), 153–175. doi:10.1007/s12553-013-0051-x
21. Jahnke-Weber, J. H. (2009). Design of decoupled clinical decision support for service-oriented architectures. International Journal of Software Engineering & Knowledge Engineering, 19(2), 159–183. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=41430638&site=ehost-live
22. Jahnke-Weber, J. H., Price, M., & McCallum, G. (2008). Making available Clinical Decision Support in Service-Oriented Architectures. Journal on Information Technology in Healthcare, 6(1), 42–54. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.5385&rep=rep1&type=pdf

23. Kamaleswaran, R., & McGregor, C. (2012). CBPsp: Complex business processes for stream processing. In 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering: Vision for a Greener Future, CCECE 2012. Montreal, QC, Canada. Retrieved from http://dx.doi.org/10.1109/CCECE.2012.6334907
24. Kameas, A., & Calemis, I. (2010). Pervasive Systems in Health Care. In H. Nakashima, H. Aghajan, & J. Augusto (Eds.), Handbook of Ambient Intelligence and Smart Environments SE - 12 (pp. 315–346). Springer US. doi:10.1007/978-0-387-93808-0_12
25. Kashyap, V. (2009). From the Bench to the Bedside: The Role of Semantic Web and Translational Medicine for Enabling the Next Generation Healthcare Enterprise. In A. Fred, J. Filipe, & H. Gamboa (Eds.), Biomedical Engineering Systems and Technologies SE - 3 (Vol. 25, pp. 35–56). Springer Berlin Heidelberg. doi:10.1007/978-3-540-92219-3_3
26. Kawamoto, K., Del Fiol, G., Orton, C., & Lobach, D. F. (2010). System-agnostic clinical decision support services: benefits and challenges for scalable decision support. The open medical informatics journal, 4, 245–54. doi:10.2174/1874431101004010245
27. Kawamoto, K., Jacobs, J., Welch, B. M., Huser, V., Paterno, M. D., Del Fiol, G., … Curtis, a C. (2012). Clinical information system services and capabilities desired for scalable, standards-based, service-oriented decision support: consensus assessment of the Health Level 7 clinical decision support Work Group. In AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium (Vol. 2012, pp. 446–55). Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3540445&tool=pmcentrez&rendertype=abstract
28. Kawamoto, K., & Lobach, D. F. (2007). Proposal for Fulfilling Strategic Objectives of the U.S. Roadmap for National Action on Decision Support through a Service-oriented Architecture Leveraging HL7 Services. Journal of the American Medical Informatics Association, 14(2), 146–155. doi:10.1197/jamia.M2298
29. Kim, J. A., Shim, Kim, S., Lee, J., Cho, I., & Kim, Y. (2011). Implementation of Guideline-Based CDSS. In Proceedings of the 2011 International Conference on Ubiquitious Computing and Multimedia Applications (UCMA 2011) (pp. 96 – 9). Los Alamitos, CA, USA. Retrieved from http://dx.doi.org/10.1109/UCMA.2011.29

30. Laleci, G. B., & Dogac, A. (2009). A Semantically Enriched Clinical Guideline Model Enabling Deployment in Heterogeneous Healthcare Environments. Information Technology in Biomedicine, IEEE Transactions on, 13(2), 263–273. doi:10.1109/TITB.2008.2010542
31. Lobach, D. F., Kawamoto, K., Anstrom, K. J., Russell, M. L., Woods, P., & Smith, D. (2007). Development, Deployment and Usability of a Point-of-Care Decision Support System for Chronic Disease Management Using the Recently-Approved HL7 Decision Support Service Standard. In K. A. Kuhn, J. R. Warren, & T. Y. Leong (Eds.), Medinfo 2007: Proceedings of the 12th World Congress on Health (Vol. 129, pp. 861–865). Retrieved from <Go to ISI>://WOS:000272064000172
32. Martínez-García, A., Moreno-Conde, A., Jódar-Sánchez, F., Leal, S., & Parra, C. (2013). Sharing clinical decisions for multimorbidity case management using social network and open-source tools. Journal of Biomedical Informatics, (0), -. doi:http://dx.doi.org/10.1016/j.jbi.2013.06.007
33. Morrison, I., & Nugrahanto, S. (2007). Decision Support With BPEL and Web Services. International Journal of Healthcare Information Systems and Informatics (IJHISI), 2(Hershey, PA, USA), 67–74. doi:10.4018/jhisi.2007040105
34. Mouttham, A., Kuziemsky, C., Langayan, D., Peyton, L., & Pereira, J. (2012). Interoperable support for collaborative, mobile, and accessible health care. Information Systems Frontiers, 14(1), 73–85. doi:10.1007/s10796-011-9296-y
35. Nakamura, M. M., Simons, W. W., Samuels, R., Daniel, J., & Mandl, K. D. (2007). Service-oriented architecture for pediatric immunization decision support. In AMIA 2007 Symposium proceedings (Vol. 14, p. 1056). Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/18694154
36. Paterno, M. D., Goldberg, H. S., Simonaitis, L., Dixon, B. E., Wright, A., Rocha, B. H., … Middleton, B. (2012). Using a service oriented architecture approach to clinical decision support: performance results from two CDS Consortium demonstrations. In AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium (Vol. 2012, pp. 690–698). Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84880810094&partnerID=40&md5=0c4b2a95e98640e314840b218c9190bd
37. Pires, D. F. . F., Teixeira, C. A. C. . A. C., Ruiz, E. E. S. . E. S. d E. S. d, Preto, R., Luis, R. W., Carlos, S., & Street, W. (2008). A UMLS interoperable solution to support collaborative diagnosis decision making over the internet. In Proceedings of

the ACM Symposium on Applied Computing (pp. 1400–1404). Fortaleza, Ceara. doi:10.1145/1363686.1364009

38. Ram, P., Berg, D., Tu, S., Mansfield, G., Ye, Q., Abarbanel, R., & Beard, N. (2004). Executing clinical practice guidelines using the SAGE execution engine. In Studies in health technology and informatics (Vol. 107, pp. 251–5). Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15360813
39. Riposan, A., Taylor, I. J., Rana, O., Owens, D. R., & Conley, E. C. (2009). The TRIACS analytical workflows platform for distributed clinical decision support. In 2009 22nd IEEE International Symposium on Computer-Based Medical Systems (CBMS) (p. 8 pp. –). Piscataway, NJ, USA. Retrieved from http://dx.doi.org/10.1109/CBMS.2009.5255422
40. Sartipi, K., Yarmand, M. H., & Down, D. G. (2007). Mined-knowledge and decision support services in electronic health. In Proceedings - ICSE 2007 Workshops: International Workshop on Systems Development in SOA Environments, SDSOA'07. Minneapolis, MN, United states. Retrieved from http://dx.doi.org/10.1109/SDSOA.2007.9
41. Steurbaut, K., Colpaert, K., Gadeyne, B., Depuydt, P., Vosters, P., Danneels, C., … De Turck, F. (2012). COSARA: Integrated Service Platform for Infection Surveillance and Antibiotic Management in the ICU. Journal of Medical Systems, 36(6), 3765–3775. doi:10.1007/s10916-012-9849-8
42. Wang, D., Peleg, M., Tu, S. W., Boxwala, A. A., Ogunyemi, O., Zeng, Q., … Shortliffe, E. H. (2004). Design and implementation of the GLIF3 guideline execution engine. Journal of Biomedical Informatics, 37(5), 305–318. doi:10.1016/j.jbi.2004.06.002
43. Wright, A., & Sittig, D. F. (2008). SANDS: A service-oriented architecture for clinical decision support in a National Health Information Network. Journal of Biomedical Informatics, 41(6), 962–981. doi:10.1016/j.jbi.2008.03.001
44. Zhu, V. J., Grannis, S. J., Rosenman, M. B., & Downs, S. M. (2009). Implementing broad scale childhood immunization decision support as a web service. In AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium (Vol. 2009, pp. 745–9). Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2815411&tool=pmcentrez&rendertype=abstract

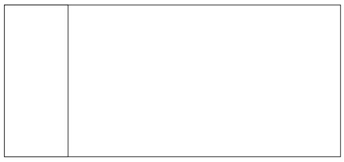
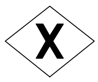
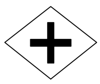
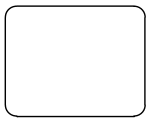
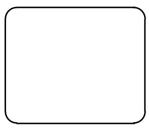
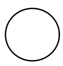
Acknowledgements

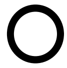
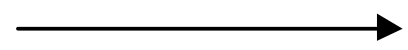
Dr. Vojtech Huser is supported by the Intramural Research Program of the National Institutes of Health Clinical Center and the National Library of Medicine.

Dr. Kensaku Kawamoto's effort for this review was supported by the University Of Utah Department Of Biomedical Informatics. Dr. Kawamoto has in the recent past or is currently serving as a consultant on CDS to the following organizations: the U.S. Office of the National Coordinator for Health IT, Partners HealthCare, RAND Corporation, ESAC, Inc., McKesson InterQual, ARUP Laboratories, Inflexxion, Inc., and Intelligent Automation, Inc. Dr. Kawamoto receives royalties for a Duke University-owned CDS technology for infectious disease management known as CustomID that he helped develop. Dr. Kawamoto was formerly a consultant for Religent, Inc. and a co-owner and consultant for Clinica Software, Inc., both of which provide commercial CDS services. Dr. Kawamoto no longer has a financial relationship with either Religent or Clinica Software.

Appendix B: BPMN Constructs used in this thesis

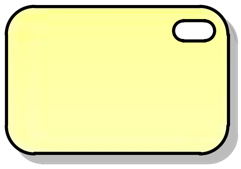
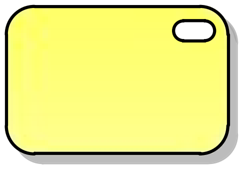
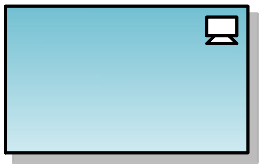
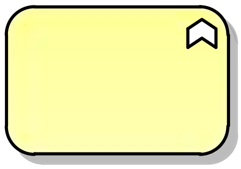
Adapted from the BPMN 2.0 specification [71].

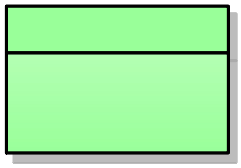
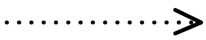
	BPMN Construct	**Description**	**Representation**
1	Pool	Graphical representation of a Participant. It may contain a process or it may not contain any details. It may represent the boundaries of the process.	
2	Lane	It is a partition of a process usually inside of a *Pool*. It may be used to define who is responsible of certain activities or to categorize the activities.	
3	Exclusive Gateway	Represents exclusive decisions and merging.	
4	Parallel Gateway	Represents forking and joining.	
5	Activity	Generic term that represents a work that is performed in a process.	
6	Task (Atomic)	Represents an atomic Activity in a business process.	
7	Start Event	Indicates where a process initiates.	

8	End Event	Indicates where a process ends.	
9	Sequence Flow	Define the order of the Activities in a process.	
10	Text Annotation	It can be used to provide additional information.	

Appendix C: ArchiMate Constructs used in this thesis

Adapted from the ArchiMate specification [142].

	ArchiMate Construct	**Description**	**Representation**
1	Application Services	Service offered for business processes	
2	Infrastructure Services	Visible behavior of both hardware and software resources	
3	Devices	Physical or hardware components	
4	Software systems	Software that is part of the infrastructure layer which can read executable files.	
5	Application Function	Behavior of the application component	
6	Application component	Self-contained unit of functionality.	

7	Application interface	route via which the application offers itself to the business or to other application (GUI, API)	
8	Data Object	An application function operates on a data object (customer record, a client database)	
9	Artifact	A byte level depiction of the application (executable files)	
10	Assignment	It may represent a relation between an active element such as Application Component and a behavioral element such as an Application Function.	
11	Used by	It defines that the element pointed by the arrow is using the element at the end without arrow head.	
12	Realization	Links logical entities such as Infrastructure Services with more concrete entities such as System Software that realize them.	
13	Access	It may indicate the access from behavioral elements (e.g., Application Function) to data objects (e.g., Data Object).	

14	Composition	Indicates that an element is composed by one or more elements.	
15	Aggregation	Indicates an element groups other elements.