UiO : Department of Informatics
University of Oslo

Interoperability for Health Information Systems

Different Architectural Approaches in Colombia

Adrian Manuel Arevalo Soria

Master's Thesis Autumn 2014

Interoperability for Health Information Systems

Adrian Manuel Arevalo Soria

1st August 2014

Every day I get in the queue (Too much, Magic Bus)

To get on the bus that takes me to you (Too much, Magic Bus)

I'm so nervous, I just sit and smile (Too much, Magic Bus)

Your house is only another mile (Too much, Magic Bus)

'Magic Bus' - The Who

UNIVERSITY OF OSLO

Abstract

MatNat
IFI

Master of Informatics: Programming and Networks

Interoperability for Health Information Systems: Different Architectural Approaches in Colombia

by Adrian Manuel Arevalo Soria

Creating interoperability for distributed heterogeneous systems is a common issue today. Practitioners literature proposes technical, syntactical and semantical solutions. On the other hand academic literature focus on organizational issues that arise during the process of making systems interoperable.

The goal of this study is to generate knowledge and identify challenges for making Health Information Systems interoperable using the HL7 Clinical Document Architecture standard (CDA). The target systems for the thesis are SIVIGILA and DHIS2 in Cauca, Colombia. Two Action Research cycles were carried out with two different architectural approaches. The first cycle was conducted in Cauca at the departmental health secretariat and the second in Norway.

The findings suggest that different strategies and architectural approaches have different challenges for making the systems interoperable. The first implementation, with a Hub and Spoke approach was unsuccessful due to lack of funds and knowledge on interoperability. The second implementation was successful with a different interoperability paradigm (SOA), architectural approach (ESB) and tools (Mule). However, challenges due to CDAs complexity and the lack of semantical interoperability codes as well as organizational challenges limited the implementation of the solution.

Acknowledgements

Isaac Newtons famous quote “If I have seen further it is by standing on the shoulders of giants” is true for my work. This master’s thesis relied on input from many people, enabling us to see further.

First of all I would like to thank my supervisors Johan Ivar Sæbø and Sahay Sundeep for guidance and valuable support through the ups and downs of the thesis, and for telling me that anything is possible.

I would like to thank the DHIS2 IT team in Cauca for all the cooperation, help and for showing me around Cauca. Thank you Knut Staring for helping me in Cauca.

I would also like to thank my family for giving me input, proof reading and believing in me when I thought there was no way to finish this work. Special thanks to Ángela Patricia Angulo for providing moral support and teaching me the spirit of Colombian dances.

Finally I would like to thank WishFire for the inspiration and artist Alfonso Espinel Rodriguez for giving me the permission to use his beautiful art in my thesis.

Contents

Abstract iii

Acknowledgements v

List of Figures xi

List of Tables xiv

List of Listings xv

Abbreviations xvii

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Relevant Literature . . . 2
- 1.3 Research Questions . . . 3
- 1.4 Research Project . . . 3
- 1.5 Structure of the thesis . . . 4
  - 1.5.1 References . . . 5

2 Literature Review 7
- 2.1 Health Information Systems . . . 7
  - 2.1.1 Defining Health Information Systems . . . 8
  - 2.1.2 Integrated Health Information System . . . 8
  - 2.1.3 Health Information System Challenges . . . 9
- 2.2 Interoperability . . . 10
  - 2.2.1 Strategies for Standardization . . . 11
  - 2.2.2 Framework of Interoperability . . . 12
  - 2.2.3 Interoperability Challenges . . . 13
    - 2.2.3.1 Organizational and Development Challenges . . . 13
    - 2.2.3.2 Semantical Challenges . . . 15
    - 2.2.3.3 Syntactical Challenges . . . 16
    - 2.2.3.4 Technical challenges . . . 17
  - 2.2.4 Interoperability Paradigms . . . 18
    - 2.2.4.1 Enterprise Application Integration (EAI) . . . 18

2.2.4.2 Service Oriented Architecture (SOA) . . . 19
2.2.5 Different Architectural Approaches . . . 20
2.2.5.1 Point to Point . . . 20
2.2.5.2 Hub and Spoke . . . 21
2.2.5.3 Enterprise Service Bus . . . 22
2.2.6 Relevant Integration Patterns . . . 23
2.3 Health Level 7 Standards . . . 25
2.3.1 HL7 v.3 RIM . . . 27
2.3.2 HL7 v3 Constrained Information Models . . . 28
2.3.3 HL7 v3 Data Types . . . 29
2.3.4 Clinical Document Architecture . . . 29
2.3.4.1 CDA Main Components . . . 30
2.3.4.2 CDA Levels . . . 31
2.3.4.3 HL7 Version 3 Critique . . . 32
2.4 Summary . . . 33

3 Methods **35**
3.1 Qualitative Research and Paradigms . . . 35
3.1.1 Qualitative Research . . . 35
3.1.2 Research Paradigms . . . 36
3.2 Action Research . . . 36
3.2.1 History . . . 36
3.2.2 Action Research, Stages and Phases . . . 37
3.2.3 Dual Masters . . . 39
3.2.4 Action Research Limitations . . . 40
3.2.5 Why Action Research . . . 41
3.3 Research Approach . . . 42
3.3.1 Research Settings . . . 43
3.3.2 My Role as a Researcher . . . 43
3.3.3 AR Cycles . . . 43
3.3.3.1 First Cycle . . . 43
3.3.3.2 Second Cycle . . . 44
3.3.4 Data Collection Methods . . . 45
3.4 Data Analysis . . . 47
3.5 Methodological Limitations . . . 47
3.6 Ethical Considerations . . . 48
3.7 Summary . . . 49

4 Interoperability Project Setting **51**
4.1 Colombia . . . 52
4.2 Health Situation in Colombia . . . 52
4.2.1 Cauca . . . 53
4.3 Health System in Colombia . . . 54
4.4 Health Information Systems in Colombia . . . 55
4.4.1 SIVIGILA . . . 55
4.4.1.1 SIVGILA the Organization . . . 55
4.4.1.2 Data Generator Units for SIVIGILA . . . 57

4.4.1.3 SIVIGILA the Software . . . 57
4.4.2 DHIS2 . . . 59
4.4.3 Other Health Information Systems . . . 60
4.5 Interoperability Project . . . 61
4.5.1 Background . . . 61
4.5.2 Interoperability Project Motivation and Collaboration . . . 63
4.5.3 Project Requirements . . . 64
4.5.4 Tracking Individual Health Events . . . 64
4.5.5 Scope of the interoperability project . . . 65
4.6 Summary . . . 65

5 First Implementation **67**
5.1 Collaboration . . . 67
5.2 Proposed Solution . . . 68
5.2.1 Implementation Scope . . . 69
5.3 Implementation Tools . . . 69
5.3.1 Mirth Connect . . . 69
5.3.2 CDA Eclipse plugin . . . 71
5.3.3 Database Tool . . . 71
5.4 Interoperability Implementation . . . 71
5.4.1 The Flat files . . . 73
5.4.2 Development of the CDA Instance . . . 73
5.4.3 Setup and Work . . . 75
5.5 Post Mirth . . . 76
5.6 Summary . . . 76

6 Second Implementation **79**
6.1 A Different Approach . . . 79
6.1.1 Collaboration . . . 80
6.2 Initial Interoperability Solution Design . . . 81
6.2.1 Finding the right Enterprise Service Bus . . . 82
6.2.2 Finding a Message Bus . . . 83
6.3 Mule Concepts . . . 83
6.4 Interoperability Solution . . . 85
6.5 Adapters in Mule . . . 85
6.5.1 SIVIGILA Mule Adapter . . . 85
6.5.2 The ActiveMQ Bus . . . 87
6.5.3 The DHIS2 Mule Adapter . . . 88
6.5.3.1 Part one . . . 88
6.5.4 Part two . . . 90
6.5.5 Part Three . . . 90
6.6 Using the CDA as a CDM . . . 91
6.7 Encountered Problems . . . 92
6.7.1 More Semantics More Problems . . . 92
6.7.2 CDA problems . . . 94
6.7.3 DHIS2 API Documentation . . . 94
6.8 CDM Conversion Details . . . 94

6.8.1 Clinical Document . . . 95
6.8.2 Participants . . . 96
6.8.3 Body . . . 101
6.9 Scope . . . 104
6.10 Summary . . . 105

7 Discussion 107
7.1 Reflections on the Technical work . . . 107
7.1.1 Architectural Approaches . . . 108
7.1.1.1 Point to Point . . . 108
7.1.1.2 Hub and Spoke . . . 109
7.1.1.3 ESB . . . 111
7.1.2 Discussion in terms Architectural Approaches . . . 112
7.1.3 CDA as CDM . . . 114
7.1.4 Using SOA Principles . . . 115
7.2 Discussion in Terms of the Interoperability Framework . . . 116
7.2.1 Organizational Level . . . 116
7.2.1.1 Different Rationalities . . . 117
7.2.2 Semantical level . . . 118
7.2.3 Syntactical Level . . . 118
7.2.4 Interaction between the Levels . . . 120
7.3 Connecting Elements with Gateways . . . 122
7.4 On research . . . 122
7.4.1 Three Challenges . . . 124
7.4.2 Fluid AR borders . . . 125
7.5 Summary . . . 125

8 Conclusion 127
8.1 Concluding Remarks . . . 127
8.2 Future Work . . . 130

A Appendix: CDA Instance for the First Implementation 133

B Appendix: CDA Derived Relevant Java Classes 135

C Appendix: Transferring Patient Information to the CDM 140

D Appendix: Transferring Patient Information from the CDM 143

E Appendix: SIVIGILA Database Model 147

F Appendix: HL7 RIM and CDA Constrained Information Models 149

Bibliography 153

List of Figures

2.1 Integrated Health Information System . . . 9
2.2 Framework for Interoperability . . . 13
2.3 Term Similarities as Semantical Challenges . . . 15
2.4 Syntacital Challenges . . . 16
2.5 Benefits of one standard . . . 17
2.6 Point to Point Approach . . . 20
2.7 Hub and Spoke Approach . . . 21
2.8 Enterprise Service Bus Approach . . . 22
2.9 Message Oriented Middleware Concepts . . . 23
2.10 Message Translator Pattern . . . 23
2.11 Channel Adapter Pattern . . . 24
2.12 Message Pattern . . . 24
2.13 Canonical Data Model Pattern . . . 24
2.14 Splitter Patterns . . . 25
2.15 Message Router Pattern . . . 25
2.16 Message Bus Pattern . . . 25
2.17 Simplified HL7 RIM . . . 27
2.18 Understanding HL7 RIM . . . 28
2.19 HL7 Data Types . . . 29
2.20 CDA Level two and three . . . 31

3.1 AR Cycle . . . 38
3.2 AR Dual Imperatives . . . 40
3.3 Project Timeline . . . 42
3.4 Hierarchy of Ethical Responsibility . . . 48

4.1 Cauca, Popayan in Colombia . . . 52
4.2 Population Pyramid . . . 53
4.3 SIVIGILA Models . . . 56
4.4 SIVIGILA Information Flow . . . 58
4.5 SIVIGILA Class Diagram . . . 59
4.6 SDSC Areas . . . 61
4.7 Involved Organizations for the Project . . . 63

5.1 Vision of a Hub and Spoke . . . 68
5.2 Mirth Connect Channel Overview . . . 70
5.3 Mirth Connect channel modes . . . 70
5.4 Sequence Diagram for the first Implementation . . . 72

5.5 SIVIGILA Basic data Form . . . 74
5.6 CDA Header and Body with standard coloring. . . . 75
5.7 CDA Participants with standard coloring. . . . 75

6.1 Architectural Approach for the Second implementation . . . 80
6.2 Solution Design for Many systems using HL7 CDA . . . 81
6.3 SIVIGILA and DHIS2 Interoperability solution Design with Scope . . . 82
6.4 Mule ESB . . . 84
6.5 Mule Message and Flow. . . . 85
6.6 Simplified Interoperability Solution Sequence Diagram for the Second Implementation . . . 86
6.7 SIVIGILA Mule Adapter Flow . . . 86
6.8 DHIS Mule Adapter Flow Part One . . . 88
6.9 DHIS Mule Adapter Part two . . . 90
6.10 MIRTH Connect Mapping . . . 90
6.11 Clinical Document . . . 95
6.12 Record Target . . . 96
6.13 Patient . . . 98
6.14 CDA Author . . . 99
6.15 Organization and Person. . . . 100
6.16 Custodian . . . 100
6.17 Body and Components . . . 101
6.18 Section and Component . . . 103
6.19 StrucDocText and StrucdocList, StrucdocItem. . . . 104
6.20 Recursive StrucDoc.Content . . . 105

7.1 Point to Point integration . . . 109
7.2 Interoperability weaknesses . . . 110
7.3 ESB Interoperability . . . 111
7.4 Interoperability Framework . . . 116
7.5 Various Interoperability Levels and their influences . . . 120
7.6 Interoperability gateway . . . 122
7.7 Action and Research Cycle within one Action Research Cycle . . . 123

A.1 CDA-view . . . 134

B.1 Author POJO Class Diagram . . . 136
B.2 Custodian POJO Class Diagram . . . 137
B.3 Record Target POJO Class Diagram . . . 138
B.4 Structured Body POJO Class Diagram . . . 139

C.1 Set Author Sequence Diagram . . . 140
C.2 Custodian Sequence Diagram . . . 141
C.3 Set Record Target Sequence Diagram. . . . 141
C.4 Set Structured Body Sequence Diagram . . . 142

D.1 Get Body for Person Translation . . . 144
D.2 Get Body from Event Translation . . . 145
D.3 Get Record Target for Person Translation . . . 146

E.1 SIVIGILA database entity relation model 148

F.1 RIM 150
F.2 CDA R-MIM 151
F.3 Screenshot of CDA HMD 152

List of Tables

4.1 Spending on Health in Colombia 53
4.2 Cauca SGSSS. 54
4.3 SIVIGILA Units and tasks (Grupo Sistemas De Información en Salud Pública SIVIGILA 2014, own translation). 58

5.1 Supported Message Standards. 71
5.2 Supported Transfer Protocols. 71

7.1 Comparing Approaches 113

Listings

2.1 CDA Main Components (Dolin et al. 2006). 30
2.2 CDA Level One Component Example (Boone 2011). 31
6.1 Line Splitter in Mule Flow. 87
6.2 Setting the Mule Flow Variable. 88
6.3 Using the Mule Flow Variable. 89
6.4 JAXB Binding One 91
6.5 Mixed Content Problems. 92
6.6 JAXB binding two and three. 92
6.7 Semantical Solution. Hard coded Data Dictionary. 93
6.8 Inject Title to CDA 95
6.9 Clinical Document part of a Marshalled CDA During Testing. 96
6.10 Record Target part of a Marshalled CDA During Testing. Values are anonymized. 97
6.11 Creating a JAXB Element. 97
6.12 Author Part of a Marshalled CDA During Testing. 99
6.13 Custodian Part of a Marshalled CDA During Testing. 101
6.14 Body Part of a Marshalled CDA During Testing. 102
6.15 Creating a JAXB Element for StrucDocList. 103
6.16 Setting StrucDocItemContent. 104

Abbreviations

AR	**A**ction **R**esearch
CDA	**C**linical **D**oucment **A**rchitecture
DANE	National Administrative Department of Statistics
DHIS2	**D**istrict **H**ealth **I**nformation **S**oftware 2
D-MIM	**D**omain **M**essage **I**nformation **M**odel
EAI	**E**nterprise **A**pplication **I**ntegration
EHR	Electronic Health Record
ESB	**E**nterprise **S**ervice **B**us
HIS	**H**ealth **I**nformation **S**ystem
HISP	**H**ealth **I**nformation **S**ystem **P**rogramme
HMIS	**H**ealth **M**anagement **I**nformation **S**ystem
HL7	**H**ealth **L**evel **7**
HMD	**H**ierarchical **M**essage **D**escription
IPS	Health Provider Institutions
INS	National Institute of Health
MT	**M**essage **T**ype
LOINC	**L**ogical **O**bservation **I**dentifiers **N**ames and **C**odes
POJO	**P**lain **O**ld **Java** **O**bject
RIPS	Individual Record of Provision for Health Services
RIM	**R**eference **I**nformation **M**odel
R-MIM	**R**efined **M**essage **I**nformation **M**odel
SDSC	Cauca's Departmental Health Secretariat
SGSSS	General Social Security System
SIVIGILA	National Public Vigilance System
SIVIGILA	National Public Vigilance Software

SISMED	Information system for Drug Prices
SNOMED	Systematized Nomenclature of Medicine
SOA	**S**ervice **O**riented **A**rchitecture
UI	Information Unit
UiO	**U**niversity **of** **O**slo
UPGD	Primary Data Generator Unit
UNM	Municipal Notification Unit
UND	Departmental Notification Unit
OID	**O**bject **I**dentifier
XSD	**X**ML **S**chema **D**efinition

Dedicated to my family

Chapter 1

Introduction

The right to health is a fundamental human right as declared by article 25 in the Universal Declaration of Human rights. Furthermore, the Alma Ata declaration, signed by more than 130 countries in 1978, stated that "Inequality in the health status of people, particularly between developed and developing countries as well as within countries, is politically, socially, and economically unacceptable" (World Health Organization 1978, p.1). However, today there are several problems that challenge the vision of health for all.

One of them is decision making based on good health information. According to Health Metrics Network (2008, p.1) health information "is an essential foundation of public health action and health systems strengthening". In addition, developing countries have challenges with their Health Information Systems (HIS) in terms of availability of information. Braa et al. (2004, p.338) argue that according to international agencies, government authorities "improved health information systems (HIS) can significantly contribute to help address health service delivery problems." Strengthening HIS can contribute to an atmosphere of reliable decision-making, rather than decision-making based on unreliable health information.

HIS may need to exchange information for various reasons, making interoperability "the ability of a system to use and share information or functionality, of another system by adhering to common standards" a relevant issue (Braa & Sahay 2012, p.59). Interoperability for HIS makes data available between systems and increases its value for the users.

The research aims to generate knowledge about interoperability for HIS by researching and implementing solutions for making two HIS in Cauca, Colombia (DHIS2 and SIVIGILA) interoperable. This chapter includes the motivation, relevant literature, research questions, research project and an overview of the chapters.

1.1 Motivation

Health Information System Programme (HISP) "is a global network established, managed and coordinated by the Department of Informatics at the University of Oslo" (*HISP* 2011). Its overall goal is to support developing countries to improve their HIS. HISP develops District Health Information System (DHIS2), a health management information system (HMIS). At the time the research began HISP was supporting a DHIS2 pilot in Cauca, Colombia at the Departmental Secretariat of health (SDSC). The pilot of DHIS2 was executed because the SDSC needed a tool to support decision making.

Information from a Colombian legacy system called SIVIGILA was needed in DHIS2 for the pilot project and for a broader implementation of DHIS2. The systems were interoperable in the pilot but it was done by a manual process. There was a need to improve the solution. This is the basis for the practical motivation.

I was introduced to DHIS2 in several university courses. The personal motivation is based on my interest in open source and working with software that is used on a global scale. I chose to work with interoperability of HIS in Cauca because I would learn about challenges and tools for a topic that is relevant not only for Colombia but also Norway. It should be mentioned that I have parents from Bolivia and know the language spoken in Colombia (Spanish). The research would increase practical knowledge but also theoretical/literature knowledge in the topic.

1.2 Relevant Literature

This research uses literature on various aspects that relates to making Health Information Systems interoperable. The challenges of making them interoperable are described by practitioners and academic literature. According to Chilundo & Aanestad (2005) the latter focuses more on organizational issues and the former on technical aspects. The research is concerned with both.

Practitioners literature describes different paradigms and architectural approaches that can be used to achieve interoperability (Erl 2004, Chappell 2004, Goel 2006, Rosen et al. 2008, Manouvrier & Ménard 2008). To implement a technical solution Hohpe & Woolf (2003) have described several relevant integration patterns.

An important pattern called canonical data model (CDM) can be used to make systems loosely coupled. (Hohpe & Woolf 2003). This research uses the Health Level 7 Clinical Document Architecture (CDA) (Benson 2012, Boone 2011) as a CDM.

Academic literature focuses more on organizational issues relating to heterogeneity of actors (Chilundo & Aanestad 2005), their rationalities (Habermas 1984, Weber 1977, Cecez-Kecmanovic et al. 2002) and power asymmetries (Webster 1995). In addition interoperability approaches entail challenges that can be described in terms of syntacital, semantical and organizational levels (Braa & Sahay 2012).

Both literatures describe challenges of making systems interoperable. They were taken into account for creating the research questions.

1.3 Research Questions

The main research question for this research is:

What are the challenges of making DHIS2 and SIVIGILA interoperable using HL7 CDA, in a landscape of distributed heterogeneous systems in Cauca, Colombia?

It incorporates the following sub questions:

- How does the architectural approach influence the interoperability solution?
- How can semantical interoperability be achieved between the two systems?
- What are the syntactical and technical challenges to make the two systems interoperable?
- How can organizational factors influence the interoperability solution?
- What strategy can be used to make them interoperable?

1.4 Research Project

In order to answer the research questions I used the research methodology Action Research (AR). It is composed of cycles that have the following phases: diagnosing, planning, action taking, evaluation and specify learning.

Action Research was carried out because the research focuses on the organizational and practical challenges. It generates knowledge through collaboration and action taking.

Thus moving away from a purely technical focus the use of AR takes into account contextual, human and organizational aspects. This research is based on two AR cycles.

The first cycle was carried out in January and February 2013 in Cauca's Departmental Secretariat of Health (SDSC). The research involved creating a proof of concept for making DHIS2 and SIVIGILA interoperable. Mirth Connect, an interoperability software, and the CDA standard was used for the proof of concept. A second AR cycle was carried out due to its failure.

The second cycle aimed at achieving interoperability with a different tool called Mule. The solution was developed in Norway with collaboration with SDSC from January to June 2014. It also involved using HL7 CDA as a CDM.

This thesis's contributions are in terms of generating knowledge on the process of making HIS interoperable in Cauca. Both the organizational and practical dimensions are included and the ways they influence each other. The practical contribution relates to knowledge based on experience with different tools and architectural approaches for creating interoperability. The organizational contribution is knowledge on the challenges that shapes and is shaped by the process of making HIS interoperable. Additionally, there are technical contributions on the use of HL7 CDA as a CDM.

1.5 Structure of the thesis

Chapter 2 "Literature" contains relevant literature on interoperability at a technical and socio-technical dimension.

Chapter 3 "Methods" provides information on the relevant methods and methodology and how they were applied.

Chapter 4 "Interoperability Project Setting" contains the setting and requirements for the research.

Chapter 5 "First Implementation" describes the first implementation using Mirth Connect and a Hub and Spoke approach.

Chapter 6 "Second Implementation" shows the second implementation with Mule and an ESB approach.

Chapter 7 "Discussion" describes the findings from the empirical work and uses the relevant literature to discuss it.

Chapter 8 "Conclusion" contains my conclusion based on the analysis and looks at the implications of the findings.

1.5.1 References

It should be noted that referencing the official CDA R2 documentation is done in by using a [§] to make it clear that the source is the official documentation.

The artworks that introduces chapter 4 ('Callejon Bohemio'), 5 ('Colombianadas'), 6 ('La Trabajosa') and 8 ('Mami Blue') are used with permission from the artist Alfonso Espinel Rodriguez.

Chapter 2

Literature Review

Abouzahr & Boerma (2005, p.582) argue that “It is not because countries are poor that they cannot afford good health information; it is because they are poor that they cannot afford to be without it.” They mention examples of how the use of data with “evidence-based decision-making” leads to better health. In addition they claim that Health Information Systems (HIS) can support public health. A common problem with HIS is fragmentation, and one proposed strategy to overcome this is to make various systems interoperable.

This chapter presents HIS and its relevant challenges in section 2.1. Strategies, a framework for interoperability and its challenges are presented in section 2.2. The latter section is relevant for showing how HIS can be made interoperable and the inextricable challenges that follows. Standards are important for making HIS interoperable. Section 2.3 describes the relevant standard for this research and relevant HL7 models.

2.1 Health Information Systems

With the term Information Systems I use a wide definition, including not only technical aspects but also organizational. This is in line with the concept of Information Infrastructures (II) (Ciborra et al. 2000). Information Infrastructure (II) theory is useful for understanding HIS interoperability challenges and strategies because the process of interoperability is not purely a technical matter. What characterizes an II according to Ciborra et al. (2000, p.60) is that it is “evolving, shared, open and heterogeneous installed base”. Additionally, an II is also enabling (Hanseth & Monteiro 1998).

II are shared because a community uses them as a "single object" (Hanseth & Monteiro 1998, p.41). They are open because there is no restriction for the number of stakeholders, users, vendors components for an II (Ciborra et al. 2000).

This leads to an II being heterogeneous due to two factors. Firstly, because II are based on interconnected smaller independent components. Secondly due to "the qualities of their constituencies": humans, organizations, institutions and components in sum making them socio-techical rather than being purely technical (Hanseth & Monteiro 1998, p.42). Thus the users, developers organizations must be active with the II for it to exist.

Finally II are based on an installed base. According to Star & Ruhleder (1996, p.133) the installed base does not grow "de novo". It is never built from scratch (Ciborra et al. 2000). New infrastructures have to be adapted and linked to the existing practices and existing infrastructures. It entails that an II struggles with the "inertia of the installed base" because it acquires its weaknesses and strengths (Star & Ruhleder 1996, p.133).

2.1.1 Defining Health Information Systems

Health Information System (HIS) is defined as "a set of components and procedures organized with the objective of generating information [that] will improve health care management decisions at all levels of the health system" (Lippeveld & Sauerborn 2000, pp. 2–3). HIS is an umbrella term incorporating various information systems. Health Management Information systems (HMIS), focus on aggregated health data. Electronic medical records (EMR), deals with patient related data. Laboratory information systems (LIS) focus on laboratory related data. Other information systems deal with logistics, finance, human resources, inventory management or are programme specific. All these information systems can be called HIS (Braa & Sahay 2012).

Health Management Information Systems (HMIS) focus on aggregated data "around the everyday provision of services" and analysis over time and geographical space (Braa & Sahay 2012, p.1). They are used for decision making on efficient allocation of resources. This research is concerned with HMIS. An integrated health information system is a form of HMIS.

2.1.2 Integrated Health Information System

To deal with the challenge of HIS fragmentation one could use a "greater degree of integration". One way to achieve integration is the use of an integrated data repository (data warehouse). A data repository is a database that has different data from various sources. Different analysis and calculations can then be made by combining data from the said sources. Such an approach can be seen in figure 2.1. (Health Metrics Network 2008, p.6)

The data repository can use several data sources to provide reports, events and alerts that can be used for evidence based decision making. The data sources can be categorized into population based and institution based. The first refers to censuses, civil registration and population surveys. The latter refers to resource records, service records and individual records. (Health Metrics Network 2008)

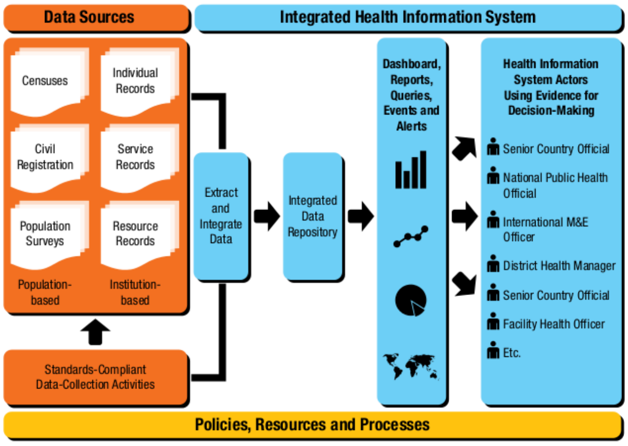


FIGURE 2.1: Integrated Health Information System (Health Metrics Network 2008).

Adaletey et al. (2013, p.3) argue that one benefit of using a data repository to integrate sub systems is due to the fact that "the integration process [can] begin without stakeholders being forced to give up their parallel systems". Other benefits include the possibility for combining data from the sources. It opens up new ways to use the information at different levels. Organization units can view and compare data from different data sources: A District can for example compare facilities in different districts based on calculations using different sources (Health Metrics Network 2008).

2.1.3 Health Information System Challenges

Health Information Systems (HIS) face various challenges such as fragmentation and low data quality. Fragmentation is based on different "administrative, economic, legal or donor pressures" (Abouzahr & Boerma 2005, p.579). It can lead to data duplication because different programs can collect the same data. Sæbø (2013, p.47) argues that fragmentation occurs due to various reasons. Different HIS have various domains and needs that creates specialized HIS: The needs for "eye surgeons are different from that of a psychiatrists, or general practitioner, and so on". In addition, uncoordinated health programmes may have their own information systems despite having similar information needs. It leads to what he calls a vicious cycle: "donor- or program-driven information systems will create a new specific system due to lack of a quality local information system" (Sæbø 2013, p.47). Consequently, fragmentation leads to poor coordination and data quality (Braa et al. 2007).

Data quality is a related HIS challenge. Shreshta & Bodart (2000) claim that the following factors influence data quality: (1) infrastructure issues (2) Lack of knowledge in terms of collecting tools (3) failure to report data (4) reporting of inaccurate data (5) intentional reporting of false data (6) errors in processing data.

Interoperability is a challenge for integrated health information systems. One can use different approaches, and paradigms to achieve interoperability. The use of standards can be used to reduce the amount of interfaces needed. The health standard that is relevant for this research is Health Level 7 Clinical Document Architecture(CDA).

2.2 Interoperability

Chilundo & Aanestad (2005) argue that the academic literature on integration differs from the practitioners literature. The latter tends to be technical oriented while the former focuses on organizational issues. Both are complementary because they give insight into the process of interoperability. This section presents practitioners and academic literature on this topic.

Interoperability is defined by Braa & Sahay (2012, p.59) as "the ability of a system to use and share information or functionality, of another system by adhering to common standards". In addition, their interoperability framework has an organizational aspect explained in section 2.2.2. Interoperability is different integration in the academic literature. Braa & Sahay (2012), Sæbø (2013), Sæbø et al. (2011) argue that integration can be defined as making systems seem as a unified whole for the end user. For them, interoperability leads to integration for HIS.

On the other hand integration is used by the practitioners literature as a broader term. Hohpe & Woolf (2003, p.5) define integration as "connecting computer systems, companies or people". From this we see that confusion can arise when using both types of literatures with their distinct definitions on integration. While integration is seen from the end user point of view in the academic literature, the practitioners definition incorporates the technical aspect of making systems integrated.

The academic definition will not be used because it is incongruent with the practitioners definition. Thus the terms integration and interoperability are interchangeable in this document. While term integration is broad I will mostly use it as a technical term.

This section presents different aspects of interoperability. Subsection 2.2.1 presents strategies for standardization and defines standards. Subsection 2.2.2 shows a framework for interoperability containing organizational, semantical and syntactical levels. Subsection 2.2.3 presents challenges with interoperability for each level. Subsection 2.2.4 describes interoperability paradigms for making systems interoperable. Common architectural approaches are presented in section 2.2.5. Finally section 2.3 presents the health standard that is relevant for the research.

An important prerequisite for interoperability is standards. However, standardization is not trivial, and I will thus start the next subsection by examining strategies for standardization, define standards and show why they are important.

2.2.1 Strategies for Standardization

There are different strategies for making HIS interoperable. The II literature shows that II can be linked by the use of gateways. Gateways can be defined as "a piece of software that links together different sub-infrastructures into an integrated one, by translating between data representations, formats and protocols" (Braa et al. 2007, p.17). They are "converters" and "operates by inputting data in one format and converting then to another" (Ciborra et al. 2000, p.69). However, this narrow definition does not fit the concepts from the practitioners literature because it excludes parts of advanced architectural approaches, making them not a part of the link[1]. I will thus use Hanseth (2001, p.1) broad definition on gateways: "a link between different elements" because its level of abstraction fits with the practitioners literature. The use of gateways makes it possible to have different formats and link them together using gateways (Hanseth 2001). This is an alternative strategy to having all elements having the same format.

However, it is also useful to look at strategies for standardization. Some of the definitions for 'standards' are different in the academic literature. For Braa & Sahay (2012) it includes data indicators, Hanseth et al. (2012, p.1) argue that standards are "one way of doing things" and that Microsoft can be seen as a standard. The presented examples of standards are too broad for my focus area, I will focus on health standards. I also extend the definition by stating that standards are a way of doing things by a recognized

[1]It excludes the Message Bus for ESB. However the narrow definition fits with the Point to Point approach (see section 2.2.5).

authority. Some example of existing health standards are HL7 v.3 CDA, CCD and HL7 v.2.

Hanseth et al. (2012) argues that there are different standardization strategies for making standards. The anticipatory standardization uses a top down strategy that is organized by a standardization body. Here standards are specified based on anticipated user requirements. It is later "hopefully, implemented in solutions which are adopted by the anticipated users" Hanseth et al. (2012, p.3).

On the other hand the integrated solutions and flexible generification strategies focus on a more pragmatic bottom up approach where user driven projects and actual use forms standards. A process of 'bricolage' or tinkering can be used to "define own messages and formats where there are no standards" (Hanseth et al. 2012, p.13).

Bricolage is defined and used differently by various academic literatures (Baker & Nelson 2005). This research is concerned with Ciborra (2002, p.49) definition as "tinkering through the combination of resources at hand. These resources become the tools and they define *in situ* the heuristic to solve the problem. [. . .] bricolage is about leveraging the world as defined by the situation"[2].

A common denominator for all strategies is that a they relay on standards. However, the process of interoperability also entails syntacital, semantical and organizational aspects. They can be expressed as levels in a framework for interoperability (Braa & Sahay 2012).

2.2.2 Framework of Interoperability

Several factors are important for achieving an interoperability solution. Figure 2.2 on the next page shows Braa & Sahay (2012) iterative framework for interoperability. It consist of three levels: organizational, semantical and syntactical. Each level has an increasing complexity being lowest for the syntactical and highest for the organizational. This framework is adapted from Carlile (2004). His framework deals with the problematic nature of knowledge as a barrier and source to innovation rather than interoperability.

The organizational level deals with how organizational processes become aligned and create the possibility for interoperability at the lower levels. This involves negotiations. Various actors such as programmes, donors, agencies and others are involved in the process of negotiation. Its outcome is agreements on how interoperability will be implemented. Section 2.2.3.1 on the following page shows challenges related to creating organizational interoperability.

[2]See (Verjans 2005) for a discussion on the implicit irony for Ciborras use of 'tinkering' for bricolage.

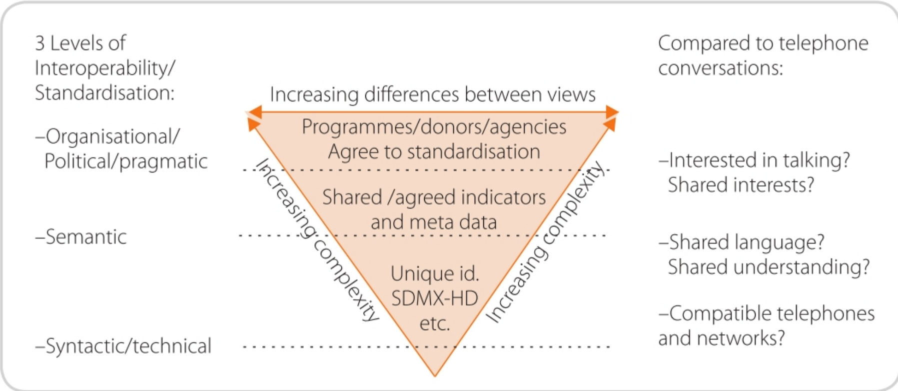


FIGURE 2.2: Framework for Interoperability(Braa & Sahay 2012).

Semantic interoperability "involves having a common understanding of what the data actually means" (Rosen et al. 2008, ch.5). Thus it involves an agreement on the meaning of data between systems that interpret it differently. Rosen et al. (2008, ch.5) highlight the importance of semantics by stating that "Without semantics, the network connectivity between consumers and services mean nothing". The agreement on the meaning of data is realized with metadata, a 'shared vocabulary' with "data dictionaries" such as ICD10 (Braa & Sahay 2012, p.68).

The syntactic level contains the grammar and the protocols to transfer data. Examples of such formats are HL7 CDA and SDMX-HD. This level solves the issues of different systems having their own formats and network protocols (Braa & Sahay 2012).

These levels are important for the development of an interoperability solution because without the syntactical/technical level there is no transfer of data. The lack of semantical agreement means that meaning of data from different systems are meaningless if transferred. Finally without organizational agreements there is no need for data to be transferred. The framework adds to our knowledge of how HIS can be made interoperable. Challenges for each level are presented in the following section.

2.2.3 Interoperability Challenges

2.2.3.1 Organizational and Development Challenges

Challenges related to interoperability at the organizational level can arise due to heterogeneity of actors and their divergent rationalities. It is challenging because it "involves multiple actors with different agendas, interests, existing technologies, and organisational cultures" (Chilundo & Aanestad 2005, p.3). Thus, achieving interoperability is not only a technical issue but incorporates organizational aspects.

As interoperability requires collaboration by different actors, its process of negotiation contains power asymmetries. The asymmetries are manifested as power imbalances, and can in turn influence the interoperability process. (Webster 1995, cited in Chilundo & Aanestad (2005)). The type of organizational relationship, from 'Hierarchial' to 'Swingle', creates different types of asymmetries (Williams 1997, cited in Chilundo & Aanestad (2005))

Heterogeneous actors rationalities influence the process. Chilundo & Aanestad (2005) present separate forms of rationalities, from different academic fields, such as 'formal', 'strategic', 'substantive', 'bounded' and 'managerial'. Weber (1977, p.64) defined formal rationality, and according to Cecez-Kecmanovic et al. (2002, p.216) it "referred primarily to the calculability of means and procedures for achieving predefined given ends".

Formal rationality was later categorized by Habermas (1984, cited in Chilundo & Aanestad (2005)) into instrumental and strategic rationality. The former is "considered when it is performed according to technical rules and when it is judged in terms of the effectiveness of intervention in a physical world" (Chilundo & Aanestad 2005, p.5). This research is concerned with instrumental rationality.

In addition to the mentioned organization challenges, the process of interoperability is also challenging because it entails to: "ensure understanding horizontally across business processes, which may be in different organizations (between domain expert and domain expert) and vertically within computer systems suppliers, between users and developers who speak different dialects" (Benson 2012). Benson (2012) argues that there are several development challenges for creating interoperability that can influence the implementation and development of the process:

- Users don't know what they want.
- Developers don't know the domain.
- Lack of communication between users and developers.
- Users do not understand the software development process.
- Users keep asking for changes.
- Developers and users fail to understand each other.

As stated previously, this is in keeping with the fact that creating interoperability is not just an technical issue but also an organizational. It is relevant for making HIS interoperable because different actors can have conflicting rationalities that can influence the interoperability process.

The organizational level only covers one part of the framework. The semantical level is interlinked with organizational aspects. I will present semantical challenges in the next subsection.

2.2.3.2 Semantical Challenges

While technical interoperability makes two systems send and receive data it is useless if they don't understand the meaning of it. Successful communication depends on sematical interoperability. A metaphor for explaining semantical challenges is typically exemplified with a telephone communication between persons (Rosen et al. 2008, Braa & Sahay 2012).

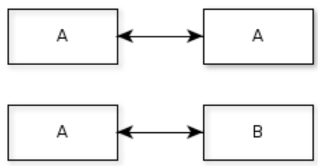
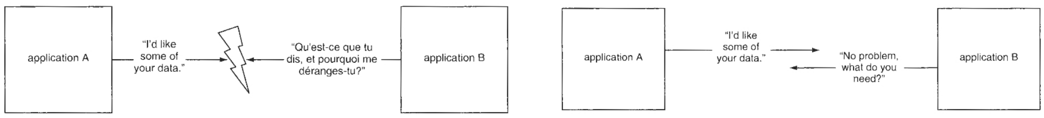


FIGURE 2.3: Term Similarities as Semantical Challenges.

Technical interoperability would make the phones connect and transfer data. The persons must then speak the same language (syntax). But to understand each other they need shared understanding of the language. (Rosen et al. 2008)

Figure 2.3 shows different scenarios where term similarities and dissimilarities gives semantical challenges. A term integration, while syntactically equal for both parties can have different meanings. For example for an IT professional and a social scientist $(A - A)$. On the other hand, syntacital different terms can have the same meaning. E.g a clinical term in English and in Latin $(A - B)$.

To summarize, effective communication is only possible if the participants/systems have a shared understanding of the exchanged data. Thus the examples presented show the need for shared understanding of meanings to make exchange useful. For HIS, this is vital if data from one system is to be understood by another. As described in the phone communication example the semantical level is dependent on the syntactical level. The next subsection presents syntacital challenges for creating interoperability.



(a) Systems with Homogeneous Formats (b) Systems with Heterogeneous Formats

FIGURE 2.4: Syntacital Challenges (Erl 2004).

2.2.3.3 Syntactical Challenges

Syntacital challenges arise due to the formats systems have in an interoperability scenario. Erl (2004) argues that interoperability challenges with homogeneous systems are less challenging than with heterogeneous systems. Interoperability of homogeneous systems can be seen in figure 2.4(b), if the systems have a common information format the effort to integrate the systems is minimal (Erl 2004).

However, figure 2.4(a) shows two heterogeneous systems that have different information formats, this setting gives rise to a more difficult interoperability scenario, according to Erl (2004). To face the challenges of having heterogeneous information formats one could use a Canonical Data Model format (CDM). Its integration pattern is presented in section 2.2.6 on page 23. This research is concerned with interoperability of heterogeneous systems.

There are two main challenges that emerges for heterogeneous systems at the syntacital level. Figure 2.5 on the next page shows how a scenario without a common standard increases the amount of work needed to integrate different systems as the number of interfaces to connect the systems increases compared to a scenario where a single standard is used (Benson 2012). The amount of interfaces needed in the former scenario can be calculated by function 2.1, where x indicates the number of systems and $f(x)$ is the number of interfaces that are required given x systems.

$$f(x) = \frac{(x^2 - x)}{2} \tag{2.1}$$

However, using a CDM in an architectural approach such as Hub and spoke or Enterprise Service Bus (ESB), that are described in section 2.2.5.3 on page 22, makes it possible to reduce the amount of interfaces. It can be reduced from the amount given function 2.1 to x interfaces for x systems (Hohpe & Woolf 2003, Benson 2012).

A CDM is thus a common format for the messages that are exchanged. According to Hohpe & Woolf (2003) it can be realized by changing the systems internal data format, using a message mapper in the system or using external message translators. The latter

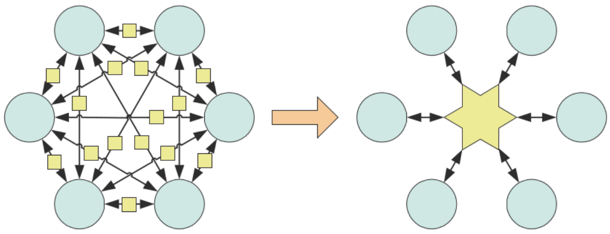


FIGURE 2.5: Benefits of one standard (Benson 2012).

is relevant for this research. Consequently, rather than having to write translations only using the applications formats, a CDM reduces the amount of interfaces needed and adding a system would just require translating the systems format to the CDM. The other systems only have to know the CDM format in their translators (Rademakers & Dirksen 2009, p.86).

On the other hand, while a solution with a CDM can reduce the amount of interfaces needed to make systems interoperable it is more complex than a solution without a it. If there are few systems that need to be made interoperable the complexity of using a CDM could potentially outweigh its benefits (Hohpe & Woolf 2003).

From this, we can see that it adds to our knowledge of making HIS interoperable. Various HIS can be heterogeneous and have different formats. The use of an intermediate 'language' or CDM is needed for reducing the amount of interfaces per system in an interoperability solution (with many systems). This level has challenges that are influenced by technical challenges. The following section presents technical challenges.

2.2.3.4 Technical challenges

Hohpe & Woolf (2003) argue that there are several technical challenges for getting distributed systems interoperable. The relevant challenges for this work are: any two systems are different and that change is inevitable.

The interoperability of two or more systems faces the challenge of heterogeneity. The difference in systems can be seen as "programming languages, operating platforms and data formats" (Hohpe & Woolf 2003, p.xxix). Change is inevitable addresses the fact that systems change over time. It is therefore important for any interoperability solution to adapt. The interoperability solution should take into account the danger of having one system change affect other systems in the solution. Therefore an interoperability solution should "minimize the dependencies from one system to another by using *loose coupling* between applications"[3] (p.xxix Hohpe & Woolf 2003, my emphasis).

Interoperable HIS solutions faces these challenges, due to different needs giving various HIS different formats. In addition, change is inevitable for HIS as updates and or new technologies emerges causing system to change. Thus without loosely coupled solutions, change in one system can influence others. This results in what Hohpe & Woolf (2003) calls 'spaghetti integration'. These challenges are the same for an interoperability solution using Service Oriented Architecture (SOA) or Enterprise Application Integration (EAI) as interoperability paradigms. The paradigms are useful for creating loosely coupled interoperable systems, they are presented in the next subsection.

2.2.4 Interoperability Paradigms

There are two main interoperability paradigms: Enterprise Application Integration (EAI) and Service Oriented Architecture (SOA), that are relevant for this research. This section presents both and shows a link between them.

2.2.4.1 Enterprise Application Integration (EAI)

The need to have distributed systems communicate within an enterprise generated interoperability between systems. Josuttis (2007, p.17) argues that 'high interoperability' is possible if the systems are able to easily connect to reach organizational goals. The demand for high interoperability gave rise Enterprise application Integration (EAI) (Manouvrier & Ménard 2008). It first appeared in works by Schulte (1997), Gilpin (1999, cited in Manouvrier & Ménard (2008)). According to Goel (2006) Hub and Spoke is an architectural approach for EAI (see section 2.2.5.2 on page 21). Manouvrier & Ménard (2008, p.23) define EAI as:

> "a collection of methods, tools, and services that work together to bring heterogeneous applications into communication, as part of the traditional, distributed or extended enterprise."

Service Oriented Architecture (SOA), defined in section 2.2.4.2 on the following page, can be seen as "an extension of EAI" (Josuttis 2007, p.296). It focuses on shared functionality following business objectives rather than the transfer of information.

[3]Loose coupling is defined by Hohpe & Woolf (2003, p.10) as a concept that "reduce[s] the assumptions two parties (components, applications, services, programs, users) make about each other when they exchange information."

2.2.4.2 Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) can be defined as a paradigm and represents a "way of thinking" (Josuttis 2007, p.12). Rosen et al. (2008) and Josuttis (2007, p.11) argues that it's challenging to define SOA because there are many different definitions. After reading several definitions on SOA (Holley & Arsanjani 2010, Josuttis 2007, Rosen et al. 2008, Manouvrier & Ménard 2008), the chosen definition for this research, for its simplicity and preciseness, is:

> "Service Oriented Architecture is a software architecture where functionality is modeled around business processes and provided as reusable services. Service Oriented Architecture also describes IT infrastructure that allows applications to exchange services and data in an interacting business process. One of the goals of SOA is loose coupling between services and between services and technology. Thus the effect of changes are kept as isolated as possible" (NIKT fagforum for arkitektur and Acondo 2008, own translation, p.12).

According to Rosen et al. (2008, ch.2), an architecture has main parts that combined can provide usefulness. He claims that the main parts for SOA are :

- **Processes**: Business functions.
- **Services**: "Modular units of business functionality". They can be integration services or business services.
- **Integration**: "Connection to and exposure of existing applications and/or data as services".
- **Existing systems**: Different systems, including legacy systems.
- **Documents**: "Units of business information".
- **Semantics**: The meaning of information.
- **Transformation**: Translating information from different formats/semantics.
- **Communications**: Different services communicate.

While EAI provides the transfer of information between systems, SOA provides *shared functionality* through services. However, SOA does not fit all interoperability situations. It is not a silver bullet and has the largest benefits where its need is based on factors such as having "heterogeneous distributed systems with different owners". In such a scenario flexibility and scalability is necessary as opposed to a a scenario with "homogeneous systems". (Josuttis 2007, p.21)

SOA and EAI use architectural approaches to achieve interoperability. This research is concerned with three approaches. The following subsection presents the main architectural approaches.

2.2.5 Different Architectural Approaches

There are different architectural approaches for achieving interoperability between systems (Lopez 2009)[4]. Three main approaches are: point to point, Hub and Spoke and Enterprise Service Bus (ESB). They have weaknesses and strengths because they are different in terms of complexity and are suitable for different interoperability scenarios.

2.2.5.1 Point to Point

The Point to Point interoperability approach links two or more systems with an interface and is suitable when there are few systems in an interoperability solution (Lopez 2009). However, if there are many systems that need to interoperate the complexity increases to what Braa & Sahay (2012) calls 'chaos' and Hohpe & Woolf (2003) calls an 'integration spaghetti'. Figure 2.6 shows a point to point approach with six systems.

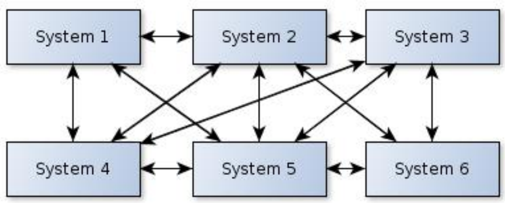


FIGURE 2.6: Point to Point interoperability (adapted from (Manouvrier & Ménard 2008)).

There is no Canonical Data Model (CDM) for this approach. Instead, each interface translates the source format to the target format. This approach leads to "tightly coupled applications and is not a scalable approach" (Liu & Özsu 2009, p.992). The consequence of a point to point approach could lead to a situation where "The system itself starts to resist change, and the result could be termed the *fossilization* of the information

[4]He uses names such as 'message server' for Hub and Spoke. I did not find any reference to a 'message server' in the technical literature but rather Hub and Spoke/message broker. He also uses 'mediator' for ESB. It was also not found in the technical literature thus I used the common names from the technical literature.

system" (Manouvrier & Ménard 2008, p.12, my emphasis). Its shortcomings results in lack of flexibility and agility because maintaining and adding systems becomes difficult. A natural solution to this problem is to have an centralized solution, such as Hub and Spoke.

2.2.5.2 Hub and Spoke

The Hub and Spoke approach consists of a central *hub* that deals with mediation, transformation, routing, and *spokes* that connect the systems to the hub (Goel 2006, Erl 2004). Figure 2.7 shows a Hub and Spoke approach with six systems. Compared to the point to point approach it only needs x interfaces for x systems with a CDM.

Goel (2006, p.3) argues that the hub provides translation from a system format to a format that other systems can understand. The centralized hub also "coordinates all communication between senders and receivers" (Josuttis 2007, p.297).

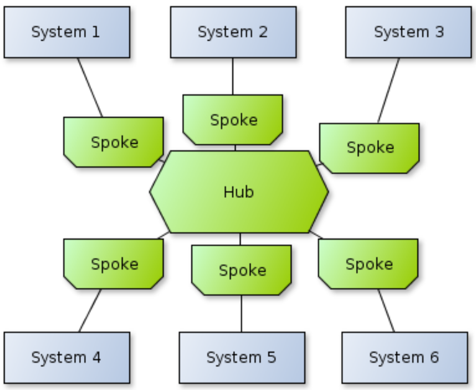


FIGURE 2.7: Hub and Spoke with Six Applications (adapted from (Goel 2006).

The drawbacks to the approach are: Scaling up the capacity of interoperability is difficult making the hub a potential bottleneck. Being a single point of failure is another drawback because it rises the solution's vulnerability (Chappell 2004, Rosen et al. 2008).

To face the challenge of having a central hub being a single point of failure one could take measures such as replication and clustering. The issues due to centralization are not applicable for a distributed ESB approach whose roots can be traced back to the Hub and Spoke approach (Rosen et al. 2008).

2.2.5.3 Enterprise Service Bus

Rosen et al. (2008) claim that an Enterprise Service Bus (ESB) has many different definitions but can be seen as an architectural pattern for a distributed infrastructure. The architectural pattern offered by distributed infrastructure services are; service location/routing directory, transactional support, transformation, mediation, specialized engines, monitoring and service security support (Rosen et al. 2008).

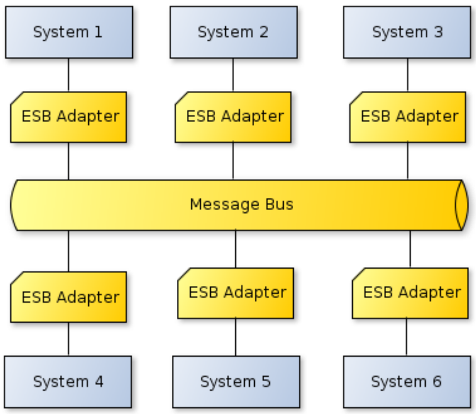
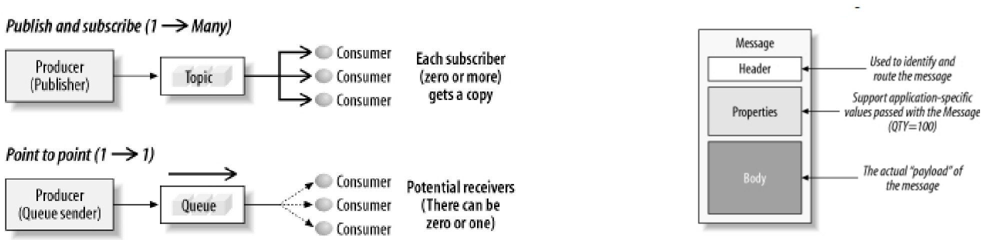


Figure 2.8: Enterprise Service Bus (adapted from Mason (2011)).

ESB products are implementations of the architectural pattern. There are many ESB products e.g : Mule, Apache ServiceMix, Microsoft BizTalk Server, Progress Sonic ESB, JBoss ESB and many more.

An ESB is an important part of SOA (Josuttis 2007) and provides "loosely coupled connectivity between service requesters and service providers in service-oriented solutions" (Liu & Özsu 2009, p.997). Figure 2.8 shows an ESB and a bus with six systems. The translation is done in the ESB adapters thus giving a distributed solution and removing the 'single point of failure' Hub and Spoke issue. Central in the figure is a message bus also known as Message Oriented Middleware (MOM).

MOM includes communication channels that enable systems to send information with messages. There are providers and consumers of messages. Figure 2.9(a) on the next page shows MOM models and figure 2.9(b) on the following page a MOM message. MOM models are *topic* and *queue*. The former is a publish subscribe domain that enables one producer to send messages to an unknown number of consumers (subscribers). The latter is a point to point domain were one producer can send messages to one consumer. Additionally, MOM messages have three elements: Body, Properties and a header. (Chappell 2004)



(a) MOM Models (b) MOM Message

FIGURE 2.9: Message Oriented Middleware Concepts (Chappell 2004).

To conclude, an ESB differs from a Hub and Spoke approach because it is distributed rather than centralized. It also reduces the amount of interfaces compared to a point to point approach to x interfaces for x systems.

In relation to HIS, the different architectural approaches can be used to make HIS interoperable. The three main approaches that are mentioned all have challenges and fit for different interoperability scenarios.

Many integration patterns can be used in the approaches. The seminal work on integration patterns, by Hohpe & Woolf (2003) presents common patterns used in interoperability solutions. Relevant patterns for this research are shown the following subsection.

2.2.6 Relevant Integration Patterns

Hohpe & Woolf (2003) work on integration patterns is highly relevant for an interoperability solution. The patterns have a name, context description, problem statement and a solution. They are important in terms of SOA because "much of this loose coupling is a result of message-based communication between services, our patterns are extremely relevant in the SOA space" (Hohpe & Woolf n.d.) Today the patterns are used in ESB products such as Mule and FuseIDE. This section presents the relevant integration patterns for the research.

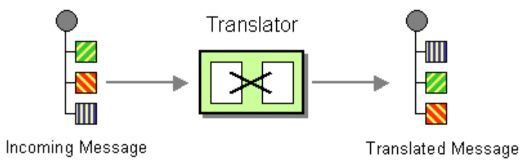


FIGURE 2.10: Message Translator (Hohpe & Woolf 2003).

The **Message Translator** shown in figure 2.10 translates an incoming message to a translated message. Different systems in an interoperability solution usually have different formats creating the need for translations. (Hohpe & Woolf 2003)

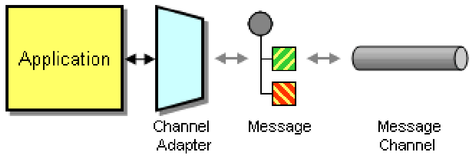


FIGURE 2.11: Channel Adapter (Hohpe & Woolf 2003)

The **Channel Adapter** shown in figure 2.11 deals with the problem of connecting a system into the message bus. The adapter can "access the application's API or data and publish messages on a channel based on this data, and that likewise can receive messages and invoke functionality inside the application" (Hohpe & Woolf 2003, p.128).

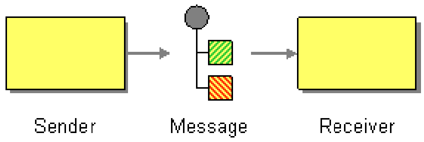


FIGURE 2.12: Message (Hohpe & Woolf 2003).

The **Message** pattern shown in figure 2.12 deals with how two or more systems exchange information with a message channel. The problem is resolved by converting the information to a message that is sent into the channel. Data transferred between 2 systems must be message. (Hohpe & Woolf 2003)

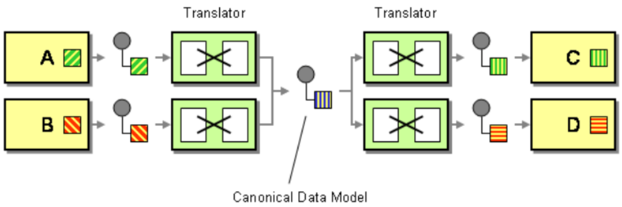


FIGURE 2.13: Canonical Data Model (Hohpe & Woolf 2003).

The **Canonical Data Model** (CDM) shown in figure 2.13 deals with the problem of minimizing "dependencies when integrating applications that use different data formats" (Hohpe & Woolf 2003, p.355). When adding new systems one only have to translate their formats to the CDM (Hohpe & Woolf 2003).

The **Splitter** pattern shown in figure 2.14 solves the problem of having a message with different elements that should be processed differently by taking a given message and splitting it into the required elements (Hohpe & Woolf 2003).

The **Message Router** shown in figure 2.15 deals with the problem of routing different messages based on a set of conditions. (Hohpe & Woolf 2003)

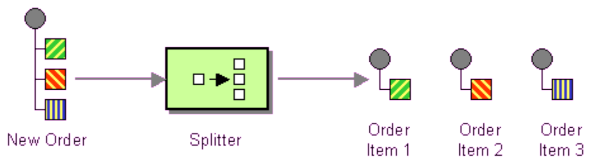


FIGURE 2.14: Splitter Pattern (Hohpe & Woolf 2003).

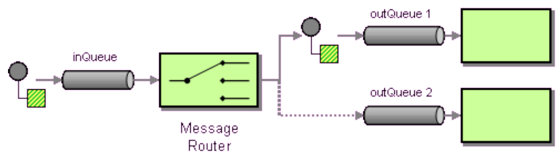


FIGURE 2.15: Message Router (Hohpe & Woolf 2003).

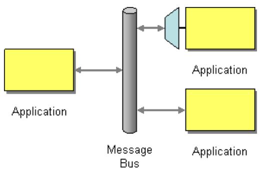


FIGURE 2.16: Message Bus (Hohpe & Woolf 2003).

Figure 2.16 shows a **Message Bus** with 3 applications. The Message Bus includes a CDM a common command set and a messaging infrastructure to allow different systems to communicate through a shared set of interfaces. A Message Bus can make applications operate independently but "work together in an unified manner" (Hohpe & Woolf 2003, p.137).

This adds to our technical knowledge on making HIS interoperable. Patterns can be used to develop the technical part of a HIS interoperability solution. This research is concerned with using interoperability patterns, approaches and elements from paradigms to make Health Information Systems (HIS) interoperable. A prerequisite for interoperability is standards. The next section presents the relevant standard for this research.

2.3 Health Level 7 Standards

Health standards can be used as a CDM to provide loose coupling in EAI or SOA implementations. The relevant standard for this research is developed by HL7 and called Clinical Document Architecture (CDA). Health Level 7 international (HL7) is an American National Standards Institute (ANSI) accredited standards developing organization. It produces standards for the health domain. The standards focuses on clinical and administrative purposes (Benson 2012). According to (Benson 2012, p.78), HL7 collaborates with

> "international (ISO TC215) and European (CEN TC251) standards development organizations, and with other specialized SDOs such as IHTSDO (SNOMED terminology) and CDISC (clinical trials) through the Joint Initiative on SDO Global Health Informatics Standardization."

The Organization has international affiliates all over the world. This research is concerned with the HL7 Colombia affiliate (*Fundación HL7 Colombia* 2014). The name Health Level Seven refers of the seventh layer of the Open Systems Interconnect (OSI) model: the application level. The level deals with "the semantics or meaning of what is exchanged" and is why the name was chosen (Benson 2012, p.79). According to the HL7 website its mission is:

> "HL7 provides standards for interoperability that improve care delivery, optimize workflow, reduce ambiguity and enhance knowledge transfer among all of our stakeholders, including health care providers, government agencies, the vendor community, fellow SDOs and patients. (*HL7 FAQs* 2014).

HL7 has developed many standards but their primary standards are:

- CDA® Release 2
- Context Management Specifications (CCOW) V 1.6
- HL7 Version 2 Product Suite
- HL7 Version 3 Product Suite
- HL7 Version 3 Standard: Structured Product Labeling. Release 4
- HL7/ASTM Implementation Guide for CDA® R2 -Continuity of Care Document (CCD®) Release 1. (*HL7 Standards* 2014)

The relevant standard for this research is the Clinical Document Architecture Release 2 (CDA). It is presented in subsection 2.3.4 on page 29. The standard is relevant in this research because it can be used as a CDM in an interoperability solution. As stated earlier the CDM is used in the ESB and Hub and Spoke approaches. CDA is based on the HL7 Reference Information Model (RIM) and uses HL7 data types. The HL7 RIM is presented in the next subsection.

2.3.1 HL7 v.3 RIM

The Clinical Document Architecture is derived from the HL7 Reference Information Model (RIM). Figure 2.17 shows six base classes of RIM. Appendix F shows a fuller model. All the other classes are derived from this model. The main classes are:

- **Entity**: An person or organization that participates in an Act.
- **Role**: Entity participants, patient, custodian etc.
- **Participation**: Subjects and objects of the clinical statements.
- **Act**: Represents clinical actions.
- **Role Link**: Link between roles
- **Act Relationship**: Links acts together.

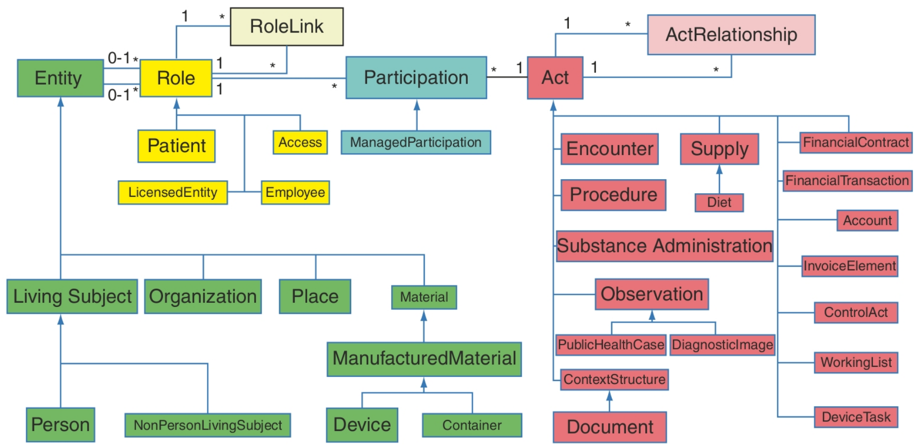


FIGURE 2.17: Simplified HL7 RIM (Boone 2011).

These classes can be separated into proper and association classes. The latter consists of ActRelationship, Participation, RoleLink, (Role Player and Role Scoper). The former consist of Act, Role, and entity. (Benson 2012)

Figure 2.18 shows how the different classes can be connected. HL7 uses its own representation of UML and the figure shows the classes in respect to their assigned colour. The association classes describe the relationship between two classes. For example, an act class can be related to itself trough the use of an association class (Act Relationship). An Act can be related to a Role through the association class Participation. A Role can be related to another Role by the Role link association class. The shapes for Association classes are arrow shaped while the the other classes are represented as squares. The bottom two sub figures can be used to make the UML diagram more compact but it will not be used in this document. (Boone 2011)

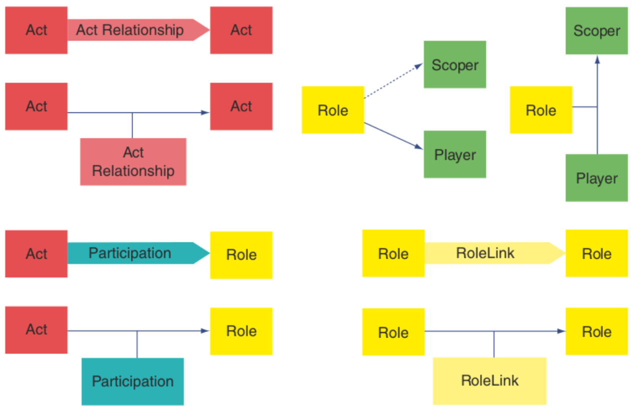


FIGURE 2.18: Understanding HL7 RIM (Boone 2011).

An important aspect to HL7 v3 is constraining models that limits optionality. The next subsection presents HL7 constrained information models.

2.3.2 HL7 v3 Constrained Information Models

Profiles or constrained information models are used for constraining a model to make it fit a use case. According to Benson (2012) HL7 v3 has a hierarchy types of constrained models:

- Domain Message Information Model (**D-MIM**)
- Refined Message Information Model (**R-MIM**)
- Hierarchical Message Description (**HMD**)
- Message Type (**MT**)

A **D-MIM** is a "general model of a domain, in HL7 notation, from which a related family of message specifications can be derived". It lacks a "hierarchical structure" and requires restriction from a **R-MIM**. **R-MIM** is a is a diagram of a message specification. It "can be expressed in a serialized format" and shown with a **HMD** in a "tabular format". Finally a **MT** can be constrained from the **HMD** or **R-MIM**. It is "a particular specification of a message which can be used in a data interchange." and they can be "exchanged as a linear string of XML and validated using an XML schema". This research uses the **R-MIM**, **HMD** and **MT** for CDA R2. (Benson 2012, p.129)

The relevant **R-MIM** and **HMD** can be seen in appendix F. They are useful for getting an overview and developing an interoperability solution with the CDA as a CDM because they provide information on different granularity levels and are complementary. In addition to these models it is also useful to get an overview of the CDA data types. Different HL7 data types are presented in the next subsection.

2.3.3 HL7 v3 Data Types

The hierarchy of data types used in HL7 v3 are shown in figure 2.19. The groups of data types are Identifiers, Boolean, Text and Multimedia, Codes, Communications and Quantities (Boone 2011). They are relevant for the CDA standard: Any interoperability implementation of the CDA as a CDM has to use some of these data types to build the CDA instance. Thus knowledge on the data type groups and the overview that presented in figure 2.19 are needed to create interoperable HIS using CDA as a CDM. The relevant standard (CDA) for this research is presented in the next subsection.

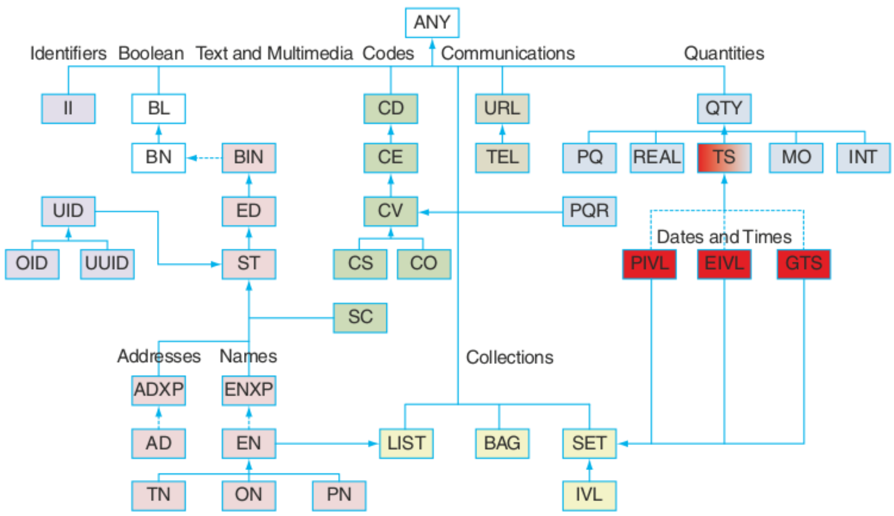


FIGURE 2.19: HL7 Data Types (Boone 2011).

2.3.4 Clinical Document Architecture

Clinical Document Architecture (CDA) is “a document markup standard that specifies the structure and semantic of a clinical document (such as a discharge summary, progressnote, procedure report) for the purpose of exchange“. It is encoded in XML and derives its “machine processable meaning from the HL7 RIM” (Dolin et al. 2006, p.31). It is also based on the HL7 v3 data types and vocabulary standards as well as on other standard vocabularies (LOINC, SNOMED) (Boone 2011).

According to Dolin et al. (2001), Boone (2011) a clinical document is defined as documentation of observations and services. They claim that a document has six characteristics: Persistence, stewardship, authentication, wholeness and human readability. CDA also has a set of main components that are described in the following section.

2.3.4.1 CDA Main Components

The CDA main components are shown in listing 2.1. However, for simplicity, it does not contain all the required components. The important components in the Clinical Document are Header and Body (Dolin et al. 2006).

```
<ClinicalDocument>
  ... CDA Header ...
  <structuredBody>
    <section>
      <text>(a.k.a "narrative block")</text>
      <observation>...</observation>
      <subsanceAdministration>
        <supply>...</supply>
      </subsanceAdministration>
      <observation>
        <externalObservation>...
        </externalObservation>
      </observation>
    </section>
    <section>
      <section>...</section>
    </section>
  </structuredBody>
</ClinicalDocument>
```

Listing 2.1: CDA Main Components (Dolin et al. 2006).

The header provides:

> "unambiguous, structured meta-data about the document itself, which can be used in document registers and data-bases to classify, find, and retrieve documents. These meta-data include information about what the document is, who created it, when, where, and for what purpose" (Benson 2012, p.150).

The Body can either be a XML or a non-XML. The latter can contain human-readable data such as text and image. The structured body contains XML encoded data (Benson 2012). Listing 2.1 shows a structured body.

The section component in listing 2.1 can contain a narrative block and entries. The narrative block is in the text element. It contains "XML markup that is similar to XHTML". While the narrative content is meant for human readability an entry encodes narrative content and “represents structured content provided for further computer processing”. (Dolin et al. 2006, p.31)

These main components are needed for a HIS interoperability solution with the CDA as a CDM. The choice of the CDA body will impact the complexity of the CDA instance because a nonXML body with text is less complex than a structuredBody with entries. In addition to the main components, the CDA can be expressed in different levels.

2.3.4.2 CDA Levels

A CDA R2 can be level one, two or three. All levels include a CDA header. In level one, “the body can be a PDF document, a jpeg image, or a text document, possibly containing simple formatting markup” (Benson 2012, p.148). Listing 2.2 shows an example of a level one CDA component with a NonXMLBody. The higher the level the higher the degree of “semantic interoperability into the exchange of the clinical documents” (Boone 2011, p.20).

```
<component>
  <nonXMLBody>
    <text mediaType='text/plain'><![CDATA[
      This is a narrative text report.
    ]]></text>
  </nonXMLBody>
</component>
```

Listing 2.2: CDA Level One Component Example (Boone 2011).

Level two is composed of an unstructured blob or a structured body with one or more sections with a given code, title and text. Its narrative block contains CDA XML encoded data. Level three introduces entries to specify text elements in a given section. It ”allows each section to include machine-processed entries at almost any level of granularity.” (Benson 2012, 148)

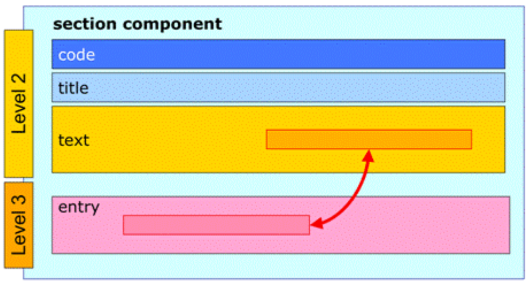


Figure 2.20: CDA Level two and three (*Body* 2014).

The difference between level two and three is seen in figure 2.20. However, there are different ’shades’ of levels. According to Boone (2011) experts have identified 12 shades.

The levels are important because they influence the complexity of the CDA for any given interoperability implementation. For example having a level one is less complex than a level three CDA.

To summarize, the CDA standard is based on HL7 v3 RIM and HL7 v3 data types. It can have different levels. In addition, CDA is implemented around the world and has generated both academical and practitioner criticism. The following section presents critique of HL7 v.3 RIM and CDA based on both literatures.

2.3.4.3 HL7 Version 3 Critique

An academic critique of HL7 v.3 is presented by Smith & Ceusters (2006). They argue that the HL7 v.3 RIM has several problems which can be categorized as problems of:

- Implementation
- Usability in specialist domains
- Technical problems
- Antiquatedness
- Scope
- HL7 business model
- HL7 governance process
- Documentation
- Learnability
- Marketing

They also claim that the RIM "is constructed on the basis of a systematically ambiguous use of terms to refer both to information objects and to corresponding real world thing and processes" (Smith & Ceusters 2006, p.141). In addition, some 'double standards' exists because there are co-existing competing conceptions for classes of the RIM. The co-existing conceptions are related to the "unsure treatment of the distinctions between information about an action on hand and this action itself on the other" (Smith & Ceusters 2006, p.135).

The literature from the practitioners critiques the CDA based on experience. Grieve (2014), a HL7 expert, editor of several key HL7 standards (v2, v3, and CDA) and a principal architect for the emerging HL7 standard FHIR, summed up lessons learned from implementing CDA in PCEHR in Australia. One of the issues was that "implementers needed to read many thousands of pages of documentation in order to implement the CDA documents correctly" and it was challenging to look for the documentation as it was spread across specifications Grieve (2014, p.54). He argues that using the HL7 RIM "presented an ongoing challenge through the project" because of its complex grammatical statements. A lesson learned is that CDA "adoption requires expertise, and policies and tools to leverage that expertise as much as possible" (Grieve 2014, p.54).

Spronk & Grieve (2008) examined XML instances of HL7's CDA R2 from 26 projects in 14 countries with a CDA schema and a MIF based validator and found common issue on: Human readability, text formatting, textual parts and corresponding entries, data types, coding issues, extensions, element sequencing and post publication infrastructural extensions.

Some issues could be found using a MIF validator such as one used in the eclipse instance editor (*Eclipse Instance Editor* 2009). The fact that 81% of the projects failed MIF based validation shows that there exists systematic challenges using CDA that should be taken into account (Spronk & Grieve 2008). Based on their findings the authors recommend that a MIF based validator should be used when working with CDA.

Thus a HIS interoperability solution with the CDA as a CDM should take into account the challenges, mentioned by both literatures, that entails working with the standard.

2.4 Summary

This chapter presented different HIS and its challenges. It also described interoperability for HIS and its challenges based on academic and practitioners literature. Challenges were presented that corresponds to the three levels of Braa & Sahay (2012) framework. HL7 CDA, the relevant health standard for this research, was also presented. It can be used as a CDM in a Hub and Spoke and ESB architectural approach.

It should be noted that the literature on ICT for development is not used in this chapter because it is not highly relevant for my research. While Colombia is a developing country my research questions does not depend on its ICT access. There are also many different ways of defining the technical concepts that were defined in this chapter making it a 'jungle' of definitions that is time consuming to navigate through. I chose the definitions based on two criteria: their relevance for my research and easiness to grasp.

Chapter 3

Methods

This chapter presents an overview of relevant methods and methodologies. It defines and explains the application of qualitative research, methodologies, data analysis and methods in terms of my research.

A practical study was done to best answer the research questions. It involves working to solve an issue of interoperability in collaboration with representatives of Cauca's Departmental Secretariat of Health (SDSC). My field work lasted from January to February 2013 where I was involved with working on solving an issue of interoperability between DHIS2 and SIVIGILA.

This chapter presents the qualitative research and paradigm (section 3.1) that was used in the research. The chosen methodology, Action Research, is described in section 3.2. Section 3.3 shows how I applied AR to execute my research. Section 3.4 presents how I analysed generated data. Section 3.5 shows the methodological limitations of my research approach. Finally section 3.6 describes ethical considerations that were relevant for my research.

3.1 Qualitative Research and Paradigms

3.1.1 Qualitative Research

As mentioned earlier (Chapter 1) the goal of this research is to find challenges of making DHIS2 and SIVIGILA interoperable. My research focuses on qualitative research to answer the research questions and find challenges for making the systems interoperable. Myers (1997) argues that qualitative research "involves the use of qualitative data, such as interviews, documents, and participant observation data, to understand and explain social phenomena". This contrasts with quantitative research that focuses on surveys, laboratory experiments, large samples and numerical methods (Myers 1997).

Thus qualitative research was used to explain the phenomenon of making DHIS2 and SIVIGILA interoperable in Cauca rather than focusing on a large scope and laboratory experiments.

3.1.2 Research Paradigms

There are different classifications for research that deal with how a researcher understands the world. These can be called research paradigms, they influence the research process in various ways because they refer to "the underlying epistemology which guides the research" (Myers 1997). Myers (1997) describes three research paradigms for qualitative research: Critical, Positivist and Interpretivist. The Interpretivist research paradigm has a philosophical base in hermenautics and phenomenology (Myers 1997). It also builds upon the assumption "that people create and associate their own subjective and intersubjective meanings as they interact with the world around them" (Orlikowski & Baroudi 1991, p.5).

The Interpretivist paradigm will be used in this research because my view of the world correlates with its epistemological foundation. Rather than believing that it is possible to obtain objective knowledge from a single reality I believe that knowledge is obtained from several realities that are social constructions. Additionally, my research will focus on the a specific context instead of focusing on generalisation.

The Interpretivist paradigm and qualitative research are relevant in combination with my chosen research methodology (Action Research). The next section shows AR's history, details and its limitations.

3.2 Action Research

3.2.1 History

Action Research (AR) is a methodology having its origins in the the field of social psychology. It was initially developed by Kurt Lewin in the 1940's and independently by the Tavstock Clinic (Baskerville 1999). However Masters (1995) argues that there was use of Action Research by others before Lewin and that the origin of AR is clouded.

One of the first articles on AR is titled 'Action Research and Minority Problems' where Lewin presented its planning, executions and reconnaissance (Lewin 1946). According to

Susman & Evered (1978, p.587) Action Research for Lewin is composed of a process that "is conceived as 'a spiral of steps, each of which is composed of a circle of planning, action, and fact-finding about the result of the action' (1946: 206)". Additionally, Adelman (1993, p.9) argues that

> "Action research for Lewin was exemplified by the discussion of problems followed by group decisions on how to proceed. Action research must include the active participation by those who have to carry out the work in the exploration of problems that they identify and anticipate. After investigation of these problems the group makes decisions, monitoring and keeping note of the consequences. Regular reviews of progress follow".

Lewin outlined phases in an iterative model that influenced later versions of Action Research and would be introduced to the field of Information Systems. Checkland developed 'soft systems methodology' in 1981. He created a link between AR and system development (Baskerville & Wood-Harper 2002).

There are many different forms of Action Research, such as (Baskerville 1999):

- ETHICS
- Cannonical Action Research
- Mulitiview
- Action Science
- Action Learning
- Soft Systems Methodology
- Participatory Action Research

3.2.2 Action Research, Stages and Phases

Action Research is a methodology that is suitable for qualitative research in the interpretive paradigm (Myers 1997). According to Avison (2002, p.19) Action Research

> "aims to improve practice through the collaborative work of researchers and practitioners. It is a synergistic relationship because practitioners, and the improvement of practice, inform research and researchers whilst, at the same time, researchers apply their theories to practice."

AR is then characterized by being problem and knowledge oriented. This duality will be described in section 3.2.3. In addition Action Research is characterized by its process that can consist of two main stages and 5 phases.

The two main stages in AR are a diagnostic and therapeutic stage. The first stage "involves a collaborative analysis of the social situation by the researcher and the subjects of the research". The second stage is action oriented and involves introducing changes into the situation. (Baskerville 1999, p.6)

The process of AR is divided into 5 phases: (1) Diagnosing (2) Action Planning (3) Action Taking (4) Evaluating and (5) Specifying Learning. Figure 3.1 shows the different phases in an AR cycle. The Diagnosis phase focuses on the "identification of the primary problems that are the underlying causes of the organization desire for change." The Action Planning phase "specifies organizational actions that should relieve or improve these primary problems" (Baskerville 1999, p.15). The Action taking phase focuses on the implementation of said planned actions. The Evaluating phase involves studying the consequences of said actions. The Specifying Learning phase is an ongoing phase were generated knowledge can impact "organizational norms", used for the potential next cycles and provide "important knowledge to the scientific community for dealing with future research settings" (Baskerville 1999, p.16). The cycle can be ended or continued by this stage. (Baskerville & Wood-Harper 2002) argues that all phases are partly shaped by the client-system infrastructure

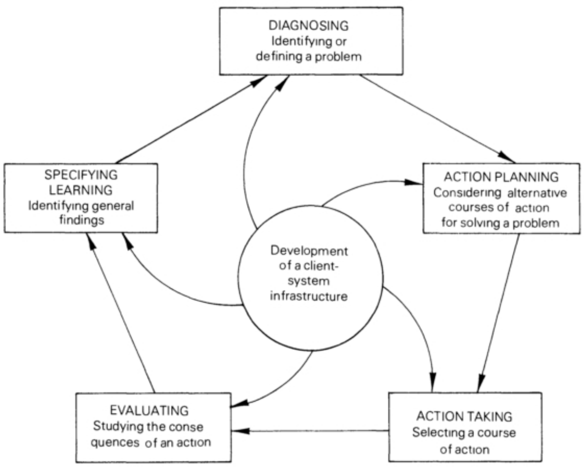


FIGURE 3.1: Action Research Cycle (Susman & Evered 1978).

Thus, AR is inextricably interlinked with the client-system infrastructure. The infrastructure is an agreement that constitutes the research environment (Baskerville 1999), it is shown in the AR cycle in figure 3.1.

Other factors also impact the process of AR. Braa et al. (2004, p.359) claim that "Action research cannot be sustainable when conceived of in a vacuum decoupled from the surrounding social, cultural, and historical context". Thus the client-system infrastructure and surrounding contexts are important factors in the process of AR and also for sustaining the process itself (Braa et al. 2004).

While the phases are conceptually distinct and separate AR is volatile in practice. Bjørn & Balka (2009, p.7) argue that

> "the borders of the cyclical processes are emergent and changing. This means that determining when one phase (such as problem diagnosis) is finished and when the next phase (such as planning intervention) begins is fluid and up for negotiation."

Thus it is important to keep the plan and description of the cyclic process phases flexible and general because quick fixes and changes can occur during its enactment (Bjørn & Balka 2009).

Validity, knowledge generation and recoverability are important dimensions of AR. (Sykes & Treleaven 2009, p.216) claim that knowledge is co-constructed in AR. It is generated by the researcher and the participants as their actions shapes and are shaped by the AR process. Validity in AR, is achieved through "knowledge created in a particular context being taken up in action" (Sykes & Treleaven 2009, p.224). The recoverability of AR is important, therefore defining the framework of Ideas, Methodology and Area of Concern important (Checkland & Holwell 1998, p.16).

3.2.3 Dual Masters

As mentioned before AR is concentrated on both intervention through action and the generation of new knowledge. McKay & Marshall (2007, p.141) argues that "the dual parallel interests in action research require researchers to conceptualise action research as being composed of two interlocking cycles of interest".

Figure 3.2 shows the dual imperatives and their respective cycles. The two interests form a separate yet inseparable cycle that, when studied and actively used in the implementation, helps avoiding the trap of over-focusing on one and forgetting about the other. The dual cycle can also help to avoid confusion and to identify "complexities and the interacting elements, and their requirements and implications" (McKay & Marshall 2007, p.148). While the dual cycle can be one of ARs strengths it can also be one of its limitations.

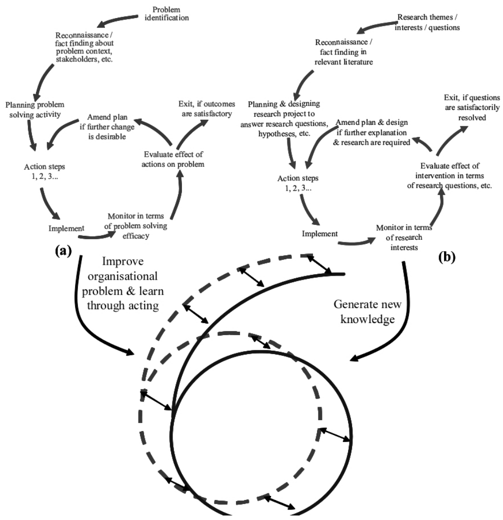


FIGURE 3.2: Dual Imperatives (McKay & Marshall 2007).

3.2.4 Action Research Limitations

A limitation to Action Research can be what Kock (2007, p.102) calls the subjectivity threat. It involves "the deep involvement of researchers with client organizations in AR studies may hinder good research by introducing personal biases in the conclusions". He also mentions the uncontrollability threat and the contingency threat. The uncontrollability threat is related to the fact that "the researcher's degree of control over the environment being studied and the research subjects is always incomplete," therefore unexpected changes to the enviroment can influence the research (Kock 2007, p. 100). The contingency threat "comes from AR's inherent obstacles to isolation of evidence related to particular effects and constructs from the contextual 'glue' in which they are naturally found" (Kock 2007, p.101).

These three problems somewhat overlap with the ones mentioned by Baskerville & Wood-Harper (2002): The lack of impartiality, lack of rigor, consulting masqueraded as research and the fact that AR is context bound rather than context free.

Additionally, the dual imperatives provide a related challenge, if one over focus on one instead of the other it can create imbalance in the research. One one hand AR can turn out to be 'consultant' like or more research oriented without a strong element of problem solving (Dickens & Watkins 1999, cited in (Davison et al. 2004)).

The context specific collaborative nature of AR means that organizational influences can have an impact the research process. While they could be positive influences, a manifestation of disagreements can put boundaries that limits the research. Some examples could include change in management or lack of resources that limits or ends the cycle of a given Action Research project.

Different interests of the participants involved in the phases can also entail challenges. Divergent interests can create unwanted tension, making control over the research a potential issue.

To summarize, there are various strengths and weaknesses to AR. The next section presents why I chose to use AR over other methodologies.

3.2.5 Why Action Research

There are several qualitative research methodologies that could be considered for my research. Myers (1997) describes three other methodologies: Ethnography, Case Studies and Grounded Theory.

Ethnography is based on having a researcher in the field of the participants or cultures, institutions that are under study. It focuses on methods such as participant observation as the researcher is immersed in the field. In this way ethnography resembles AR, the two methodologies "engage both with and in the meaning making of the participants and particular phenomena under investigation" (Sykes & Treleaven 2009, p.218). However AR focuses more on having the researcher in the foreground and participating in action taking, something that ethnography lacks (Sykes & Treleaven 2009).

On the other, hand Grounded Theory "seeks to develop theory that is grounded in data systematically gathered and analyzed" (Myers 1997). It focuses on data generation methods such as interviews and uses a considerable amount of time in its data analysis phase where the generation of inductive theory is continuously in an interplay with data generation. It also lacks the action taking aspect that Action Research has.

Case studies are characterised by an investigation of a phenomenon with boundaries in a real life context (Myers 1997). AR and Case Studies both entail generating context-bound knowledge. However AR deals with the dual imperative: problems solving and research. Thus the researcher has a more active role in AR compared to Case Studies. In addition, Case Study findings are for an academic audience while AR have an "obligation to feed data back into the community with which they collaborated when identifying and solving a practical problem" (Blichfeldt & Andersen 2006, p.4).

Thus Action Research differentiates itself from the other methodologies because it is based on the study of actions and consists of its five phases. This is why it was chosen as a methodology to best generate data for analysis and to form conclusions.

AR focuses then on collaboration, action taking and studying the effects of the action taking (Baskerville & Wood-Harper 2002). The implementation of AR could enable a process where knowledge about a specific problem in Cauca and knowledge based on problem solving would be generated. Thus the research would show context specific information about challenges for creating interoperability that could provide answers to the research questions that were shown in Chapter 1.

This research contains the action of making SIVIGILA and DHIS2 interoperable to solve a problem in Cauca. It is done in two AR cycles that involves different technical approaches, both using HL7 CDA. By conducting AR I can study the effects and discover challenges of making the systems interoperable.

To summarize, AR was chosen because it is an action oriented methodology. My AR approach used selected research methods. They and the research setting are presented in the next section.

3.3 Research Approach

The research approach includes research settings, methods for data generation and data analysis. Figure 3.3 shows a timeline with the two AR cycles and other projects at the SDSC that influenced or was influenced by the cycles[1]. The Pilot project had produced knowledge that influenced the first AR Cycle. The latter influenced the second cycle and the wider DHIS2 interoperability solution by the SDSC IT team. Finally, the second implementation influenced the wider implementation. My participation was limited to the AR cycles. Both cycles had different settings, thus the next subsection presents the research settings.

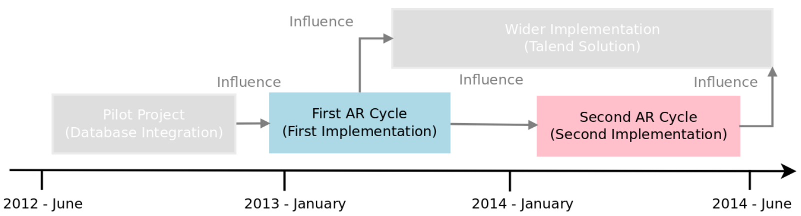


Figure 3.3: Project Timeline.

[1]The Pilot project is described in chapter 4. See chapter 5 for empirical work done in the first AR cycle and the wider implementation. Chapter 6 for empirical work done in the second cycle.

3.3.1 Research Settings

The research was performed in two consecutive cycles, both with different research settings. The first cycle had Cauca's Departmental Secretariat of Health (SDSC) as a research site where I collaborated with the IT team in charge of implementing DHIS2. I also collaborated with a professor from the University of Cauca.

The research site for the second cycle was Norway. Collaboration was mainly with the same IT team through Skype, chat and email. However, the solution was developed from the University of Oslo but relied on contact with the IT team for the five AR phases.

3.3.2 My Role as a Researcher

My role in SDSC was to participate on behalf of the University of Oslo in a development and research team at the SDSC. My role was to come up with a proof of concept for making SVIGILA and DHIS2 interoperable and provide a general skeleton for interoperability between HIS in the Cauca department. Thus I became part the the IT team working on DHIS2 in Cauca while retaining my researcher obligations.

3.3.3 AR Cycles

Two AR cycles were executed during my research. The second cycle was a direct result of the first. It was done based on the lessons learned from the first one. The two cycles are described below with their respective phases.

3.3.3.1 First Cycle

The first cycle was influenced by the knowledge gained from the Pilot Project implemented prior to its undertaking. The cycle was carried out in Colombia in the beginning of my research. It included the following AR phases:

Diagnose$_1$: Data from SIVIGILA was needed to be transferred to DHIS2 in a automatic manner. The data was transferred in a manual procedure. Additionally there was a need to integrate HIS from the health service providers in Cauca and SDSC.

Plan$_1$:

- Create an interoperability proof of concept with MIRTH Connect with HL7 CDA as a CDM.

- Connect MIRTH to SIVIGILA and DHIS2.
- Create an example CDA.
- Send data using MIRTH Connect, CDA and insert it into DHIS2.

Action$_1$: A proof of concept was developed using Mirth Connect and CDA.

Evaluation$_1$: Involved weekly meetings with the professor from University of Cauca and a SDSC worker. Work advances were evaluated.

Specify Learning$_1$: It was done during the meetings and during conversations with the SDSC IT team. It was used for basis for the next cycle.

Therefore the first cycle influenced the second one. It also influenced the Wider Interoperability solution the SDSC worked on after the cycle's end.

3.3.3.2 Second Cycle

A second AR cycle began in February 2014 where the experiences from the first cycle shaped the second one.

Diagnose$_2$: A new approach was needed to solve both problems of general interoperability and SIVIGILA to DHIS2 interoperability.

Plan$_2$: Find a different tool and approach to solve the problems at hand. Develop a new proof of concept.

Action$_2$: Actions consisted of implementing the solution by finding the correct tools and implementing the solution with CDA as a CDM.

Evaluation$_2$: It was done by myself but also by communicating with the IT team in Cauca, discussing the results of actions and sharing parts of the code for my solution.

Specify Learning$_2$: It was done in collaboration with the IT team in Cauca using Skype and emails. The knowledge of DHIS2 API was disseminated across the team. They began using it separately. Knowledge on CDA difficulties and organizational deficiencies were also discussed with the team.

Therefore this cycle influenced the wider implementation. The two cycles used different methods to achieve their objectives. They are mentioned below in the next subsection.

3.3.4 Data Collection Methods

The following methods were used to generate data: Interview, participant observation, document analysis, fieldnotes, and solution implementation. They were used as complimentary methods to best find information that could be used in my analysis to answer the research questions.

Interview and Conversations

There are many types of interviews, the one I focus on is the qualitative research interview. It "attempts to understand the world from the subject's point of view, to unfold the meaning of their experiences, to uncover their lived world prior to scientific explanations" (Kvale & Brinkmann 2009, p.1). An interview has different styles or techniques that vary between being formal and structured to being informal and unstructured similar to a conversation (Madden 2010).

Interviews were done in the office of SDSC. The main interviewees were two IT workers in the SDSC working with implementing DHIS2. The interviews covered broad themes and recording was done by writing down important keywords and ideas on a laptop. Conversations were done while developing the solution with the IT team in Cauca.

'Remote' interviews were performed using Skype in the second cycle. A member of the IT team was the interview. Two interviews were recorded and transcribed using a software called Transcribe.

Email

Emails were used to generate data. The main email correspondents were a SIVIGILA IT worker at SDSC, two SDSC IT team members, a University of Cauca professor, a DHIS2 developer and Grahame Grieve, a HL7 expert, editor of several key HL7 standards (v2, v3, and CDA) and a principal architect for the HL7 standard FHIR. One of the IT workers had a mandate to respond on behalf of the SDSC head office. Emails were also used to communicate with the DHIS2 developer community.

Participant Observation

Observation is a way to understand human groups by positioning the researcher in the naturally occurring settings. An observation can be either passive or participant. Participant observation is, according to Madden (2010) an observation where the researcher gets involved with the people that are being observed and becomes a part of their context. Thus it becomes a whole body experience were the body becomes an organic recording device in terms of recording the observation (Madden 2010).

I participated and observed the DHIS2 academy, developing interoperability solution with the IT-team, and a briefing on DHIS2. I used participant observation to observe the process of transferring data from SIVIGILA to DHIS2 using scripts.

Implementing a solution

I collaborated in the development of an interoperability solution following a requirements list. It entailed coding, use of tools and working with several respondents in Cauca including the IT team and the University of Cauca professor.

Fieldnotes

There are different forms of fieldnotes, Madden (2010) defines two types; 'participatory fieldnotes' and 'consolidated fieldnotes'. The first type is taken while participating, in the "hurly-burly of active fieldwork". The second type is "taken at the end of the day's work or sometime soon after an event, which expand the description and might have a more reflective and or analytical tone". (Madden 2010, p.123)

I took participatory fieldnotes to record important events while they occurred, such as: Solution implementation issues, system settings, configuration items and passwords, with a software called redNotebook. A regular notebook was used when it was more convenient, e.g to make diagrams. Consolidated fieldnotes were taken at the end of the day where I reflected on important events that occurred. A backup was taken on a regular basis of the fieldnotes saved in redNotebook.

Document Analysis

Document analysis generates relevant research topics according to relevant documents. Documents were analysed to generate data about several topics that are related to answering the research questions. The documents that were analysed used were:

- User Manuals of SIVIGILA
- Wiki on MIRTH Connect
- Presidental Decree on SIVIGILA
- HL7 CDA tutorial and other documents on CDA such as Implementation Guides and official CDA documentation.
- DHIS2 Requirement document written by the SDSC
- Documents on the health situation in Cauca and Colombia.
- Web pages that contains information about the SDSC, SIVIGILA and other HIS pages.

To summarize, the different methods were used to generate data in the research settings in two consecutive AR cycles. The next section presents how data was analysed.

3.4 Data Analysis

The qualitative data was analyzed in several steps. The first included organizing post fieldwork data and data from the fieldwork. Then, codes were used and indexed. From these codes themes were identified and finally structures of relationships between themes formed explanations that helped answer the research questions.

Data from the first cycle was analysed after its ending. On the other hand, data from the second cycle was analysed in and after it's execution. I analysed transcribed Skype calls, and emails. The second solution was also analysed as it was developed to find explanations. The AR approach was not without limitations, the next section presents methodological limitations encountered during my research.

3.5 Methodological Limitations

Methodological limitations were imposed by factors such as time, language and knowledge. I spent six weeks in Colombia for the first cycle. Understanding the health information systems, implementing a interoperability solution takes time and six weeks is little time and influenced the amount of data generation.

As stated in section 3.2 it is important to have a framework of ideas, methodology and the area of concern when conducting an Action Research (Checkland & Holwell 1998). I did my field work at the very beginning of my master thesis, before I had taken courses on Health Management Information Systems and Research Methodology, before I had time to gather knowledge about interoperability and with no time for preparation. It meant that data generation was influenced by the lack of knowledge about HISs, interoperability and Action Research. Thus the recoverability for the first cycle is low, due to the missing knowledge. It should be noted that I had more knowledge on AR, HIS and interoperability for the second cycle.

Language was another limitation. The Spanish spoken in Colombia is somewhat different from the Spanish I know (Bolivian). Some words are different and Colombians talk faster. This made it slightly difficult to understand the language in the beginning. It also took time to get used to technical terms in Spanish. Additionally, writing Spanish takes more time for me than writing English. When recording interviews I found it easier to write down some information in English, however when doing so information can get lost. Using audio recording and transcribing the interview instead of using a laptop would have made it more efficient.

An example of language difficulties that arose was the use of the term “migración” or to migrate in relation to data. In English it means a one time transfer of data from system A to B. However in Colombia, they used it to ’transfer’ data periodically.

Another limitation was having diverse sources. Theoretical papers, practitioner-oriented literature and people working in different countries in different languages used different names for the same architectural approaches concepts. There are many definitions for what an ESB is as mentioned in chapter 2. This is also true for Hub and Spoke; it could be called a message server (Lopez 2009), message broker (Hohpe & Woolf 2003) and was refereed to once in email communication as an interface server. In addition, Mirth Connect can be used as a Hub and Spoke but market itself as an interface integration engine. Thus the diverse sources own definitions on concepts made it challenging and time consuming to process.

The research dealt with sensitive data that made me take several ethical considerations. The following section shows ethical considerations that were made for this research.

3.6 Ethical Considerations

Madden (2010) claims that there is a hierarchy of responsibility when it comes to being ethical, in terms of research. Figure 3.4 shows the levels of hierarchy: The rights of the participants comes first, then the researcher and finally the discipline itself. However if a researcher encounters a situation so grave that the research should be stopped due to ethical concerns one might use a response with the Universal Human Rights (Madden 2010, p.91). Issues of ethical concern can arise in many stages during the research.

FIGURE 3.4: Hierarchy of Ethical Responsibility.

Ethical issues regarding interviews can "arise particularly because of the complexities of researching private lives and placing accounts in the public arena" (Kvale & Brinkmann 2009, p.63). It is therefore important to consider the ethical dimension of an interview. All participants of the interviews were given an explanation on how the data was going to be used. Confidentiality of the respondents is used except for one HL7 expert, because I got his consent to use his full name.

In addition, there is an issue about ethical consideration when using sensitive data. I was given access to files from SIVIGILA and DHIS2 databases that contained patient data. I was careful when handling such data and it was not shown to others. It meant that sharing my technical work with others was limited by ethical considerations.

3.7 Summary

This chapter presented the choice of research, paradigm, methodology and methods. Qualitative research was chosen to best answer the research questions with an Interpretivist paradigm. The relevant methodology was AR. It has limitations and is different from other research methodologies. Two cycles of AR were executed and used with various methods.

My research had limitations due to knowledge, time constrains and language. Finally, there were ethical consideration taken due to working with sensitive data. A part of the generated data from the research approach is shown in the following chapter.

Chapter 4

Interoperability Project Setting

My work covers an interoperability project to develop proof of concepts. This chapter contains background information for the project. Colombia is broadly covered in section 4.1. Its health situation is described in section 4.2. Section 4.3 shows the health system in Colombia. Section 4.4 presents various HIS in Colombia. Finally section 4.5 shows the background and requirements for the interoperability project. This chapter presents the background setting for two proof of concepts that were the result of two AR cycles are shown in in chapter 5 and chapter 6.

4.1 Colombia

Colombia is a South American country. It has a population number of 47.703.725 (*Poblacion Colombia Ahora* n.d.).Its capital is Bogota and the main language is Spanish. Figure 4.1 shows Colombia in South America, the department Cauca and its capital Popayan. Colombia is a republic that is divided into 32 departments.

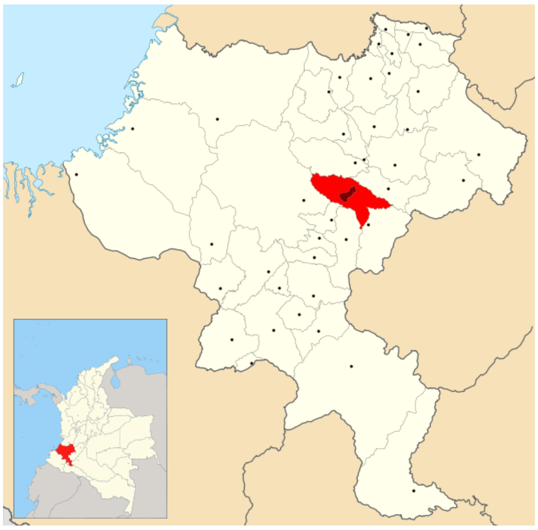


FIGURE 4.1: Cauca, Popayan in Colombia (Wikipedia).

According to the CIA factbook the population have mestizo 58%, white 20%, mulatto 14%, black 4%, mixed black-Amerindian 3% and Amerindian 1%. 90% of the population are Roman Catholic making it the main religion in Colombia. (*The World Factbook* 2013)

Figure 4.2 shows a population pyramid for Colombia (*The World Factbook* 2013). The pyramid shows that the population for ages 0-14 years: is 25.8% (male 6,032,725/female 5,755,437) and for the ages 25-54 years: 41.5% (male 9,376,745/female 9,597,744). 65 years and over: 6.5% (male 1,242,980/female 1,728,922).

4.2 Health Situation in Colombia

This section presents the health situation in Colombia and Cauca. Colombia has about 1.5 medics per 1.000 persons. Maternal mortality per 100.000 live births is 72.9. The literacy is 93.2% as in 2009. Life expectancy is 73.4 years. In the last decade there is a 100% coverage of priamry and secondary school education. The persons living under poverty was reduced between 2005 and 2009 from 50.3% to 45.5%. At the same time extreme poverty increased by 0.7% to 16.4%. The Genera Social Security System

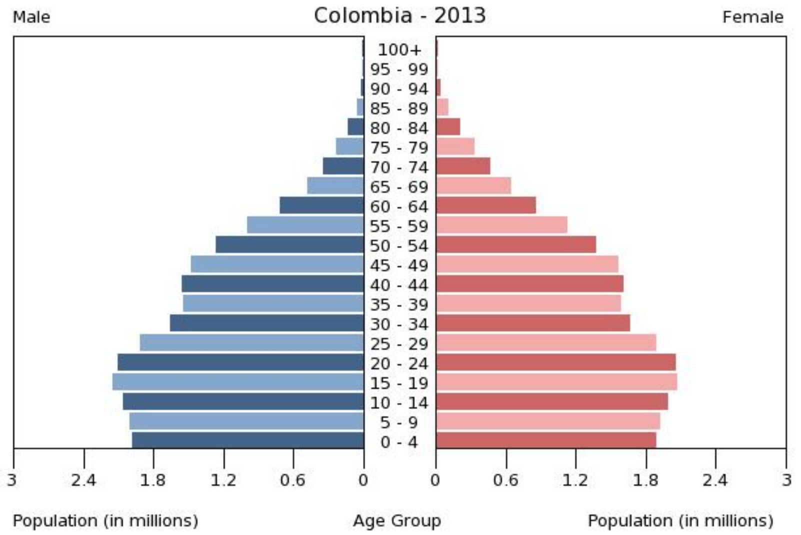


FIGURE 4.2: Population pyramid for Colombia (*The World Factbook* 2013).

(SGSSS) covers a large segment of the population (91.1%). The different levels of system will be described later in the chapter. (Pan American Health Organization 2012)

Table 4.1 shows the spending on the health sector in relation with the gross domestic product (GDP).The total spending on health is 6.4%, the most spending comes from the Public (5.4%) and the private spending is 1%. The health spending per capita in Colombia is 323$ (2009) (Minisetrio de Salud y Protección Social 2011). My research was done in the Cauca department.

Sector	Spending in %
Public	5.4
Private	1.0
Total	6.4

TABLE 4.1: Spending on Health in Colombia

4.2.1 Cauca

Cauca is a Colombian department that covers 29,308 km², has 38 municipalities and 99 districsts. The department has a population of 1.308.090 and its capital city is Popayan, has 265.881 persons. Cauca has 517.041 persons that are classified as urban population. Rural population is 801.942 and in 2010 the annual growth rate for the population was 1.18. Life expectancy is 70.8 years. (Echavarría & Granda 2010)

A number of people are displaced by violence because of the civil war in Colombia. In Cauca 2009 a total of 102.045 were displaced by violence. The public spending on health per capita in Cauca is $116 which is much lower than the national average. Cauca has 8.4% of analfabetism (for persons above 15 years) (Echavarría & Granda 2010). Figure 4.2 shows the distribution of persons and types of affiliations to SGSSS in Cauca (Echavarría & Granda 2010).

Type of Social Security	Number of Persons	% of population
Subsidised	856.403	73.1
Contributive	237.292	8.5
Non affiliated to SGSSS	213.395	18.3

Table 4.2: Cauca SGSSS.

4.3 Health System in Colombia

Colombia's health system can be divided into state agencies, insurance companies and service providers according to Boluda (2013). The State agencies are: The Ministry of Health and Social Protection, the National Health Superintendence and the National Institute of Health (INS).

The Health providers (IPS) are hospitals, health clinics, laboratories that provide health services to the population. Human resources in this aspect are professional health personnel and transport personnel that are involved in health providing. They are classified in 4 levels: (1) Institutions that provide low level complexity (medical consultation, odontology, emergency care and promotion and prevention activities). (2) Institutions that provide medium complex services (internal medicine, orthopedics, obstetrics, general surgery, pediatrics, emergency 24 hours). (3) Institutions that provide high complexity services (neurosurgery, vascular surgery, pulmonology, dermatology, Care Units (UCI). (4) The fourth level covers support for treating difficult chronic treatment such as HIV, cancer, etc. (Castrillón et al. 2012)

The health insurance providers are private entities that provide insurance to the population. The health promoting entities (EPS in spanish) organize health services. There are two health regimes in the system (SGSSS): contributory and subsidized. The contributory is for salaried workers, pensioners and independent workers with incomes above the minimum salary). 44% of the population is covered by 22 EPSes. The Health insurance for the poor and unemployed is called Subsidized regimen and is funded by taxes and other financial sources. 51% of the population uses this regimen. 96% of the population is nominally covered by both. (Francisco & Yepes 2012)

4.4 Health Information Systems in Colombia

The Colombian health systems has several health information systems. They support different areas. They are: SIVIGILA, DHIS2, RUAF, SISPRO, DANE, RIPS, SISMED, health programme specific information systems and others.

4.4.1 SIVIGILA

SIVIGILA is both an organization and software, taken together they form a HIS. They will be presented to show their differences and how they are dependent. The use of the term SIVIGILA in other chapters will refer to the software when discussing technical interoperability.

4.4.1.1 SIVGILA the Organization

SIVIGILA, the organization for public health surveillance is defined in article 3 in the Decree 3518 from 2006 that regulates and creates SIVGILA as:

> “Set of users, rules, procedures, resources (financial and technical) and human talent organized for the collection, analysis, interpretation, updating, and systematic dissemination and timely assessment of health event information for action orientation” (Uribe Velez 2006, own translation).

According to article 6 in the decree 3518 of 2006, the entities responsible for implementing and developing SIVIGILA are: the Ministry of Health and Social Protection, the National Institutes of Health (INS) and the institutes of vigilance for Food and Drugs (INVMA), Departmental, Districts and Municipals health Directorates, Notification Units and UPGDs. They have specific functions in SIVIGILA as described in articles: 7–14.

SIVIGILAs goals are as follows:

1. “Estimate the magnitude of the events of interest in public health.”
2. “Detect changes in patterns of occurrence, distribution and spread of events under surveillance in public health.”
3. “Detect outbreaks and epidemics and target specific control actions.”

4. “Identify the risk factors and protective factors related to events interest in health and population groups exposed to these factors.”
5. “Identify needs of epidemiological research.”
6. “Provide health planning and the definition of prevention and control.”
7. “To facilitate the monitoring and evaluation of health interventions.”
8. “To guide actions to improve the quality of health services.”
9. “Direct the formulation of public health policies.” (Uribe Velez 2006, Article 4, own translation)

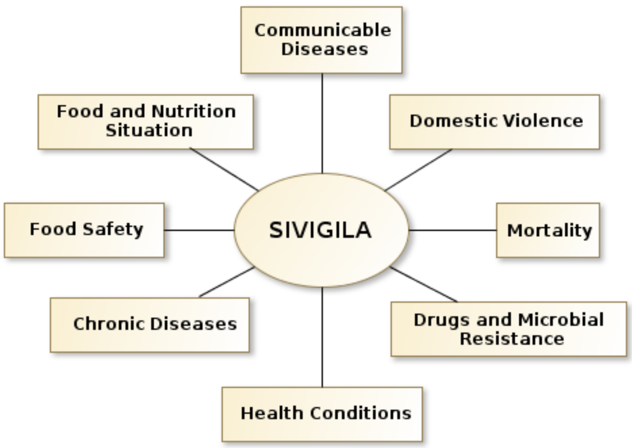


FIGURE 4.3: SIVIGILA Models.

There are 8 models for public health monitoring in SIVIGILA (Bolaños & Muñoz 2009). A model is defined in article 3 as a “conceptual construction ordering the aspects with which to cover specific problems requiring monitoring by the system” (Uribe Velez 2006). The model allows the obtension of complete information about group of interests. Figure 4.3 shows different models that SIVIGILA covers. They are: Communicable diseases, Domestic Violence, Mortality, Drugs and Microbial Resistance, Health Conditions, Chronic Diseases and Food Safety, Food and Nutrition Situation. (Bolaños & Muñoz 2009, own translation)

Health events are the set of events or circumstances which may alter or affect the health status of a community and which are the subject of public health surveillance according to article 3.

The events are collected through standardized forms that are defined by INS. These data forms for the events can have a side A and side B. The formats can be accessed online from INS website (Bolaños & Muñoz 2009, *Fichas de Notificación* n.d.).

4.4.1.2 Data Generator Units for SIVIGILA

SIVIGILA has several data generation units that are defined as UPGD, UNM, UND, and UNN:

UPGD is the primary generator of data, it covers all the health information centres, IPS, health offices or specialised units, laboratory and microscope offices. They report in general to UNM and in some occasions to UND. Cobo & Garcia (2012)

UNM is the Municipal Notification Unit that covers the Health Secretariats and the municipalities. The units are in charge of collection, verification and analysis of public health events at the municipal level. Cobo & Garcia (2012)

UND is the Departmental Notification Unit. It covers the Departmental Secretariats of Health and is in charge of the collection and analysis of public health events in the Departmental level. Cobo & Garcia (2012)

UNN is the National Notification Unit in Colombia that represents the National Institute of Health in terms of SIVIGILA. The Ministry of Health can be seen as an entity that is in charge of collection and analysis of health information. Cobo & Garcia (2012)

4.4.1.3 SIVIGILA the Software

According to article 21 in the decree of 2006, that creates and regulates SIVIGILA, SIVIGILA uses an information system "For purposes of ensuring the standardized operation of informatics processes" (Uribe Velez 2006, own translation). The Ministry of Health is responsible for the definition and regulation of the information system, establish the "institutional internal and external mechanisms for collection, transfer, update, validation, organization, disposal and administration of data that is applied in all the different levels of the system for each process related with public vigilance in public health of one event or a group of events" (Uribe Velez 2006, own translation),

The information system currently used is called SIVIGILA and was first released in 2006. INS is responsible for its planning, development and implementation (Bolaños & Muñoz 2009). The information flow and the operation tasks in SIVIGILA with the different actors and their corresponding functions are shown in figure 4.4. The Different actors are defined in section 4.4.1.2. Actors have different activities for SIVIGILA. Activities for U.I, UPGD, UNM, UND and UN are shown in table 4.3 on the following page.

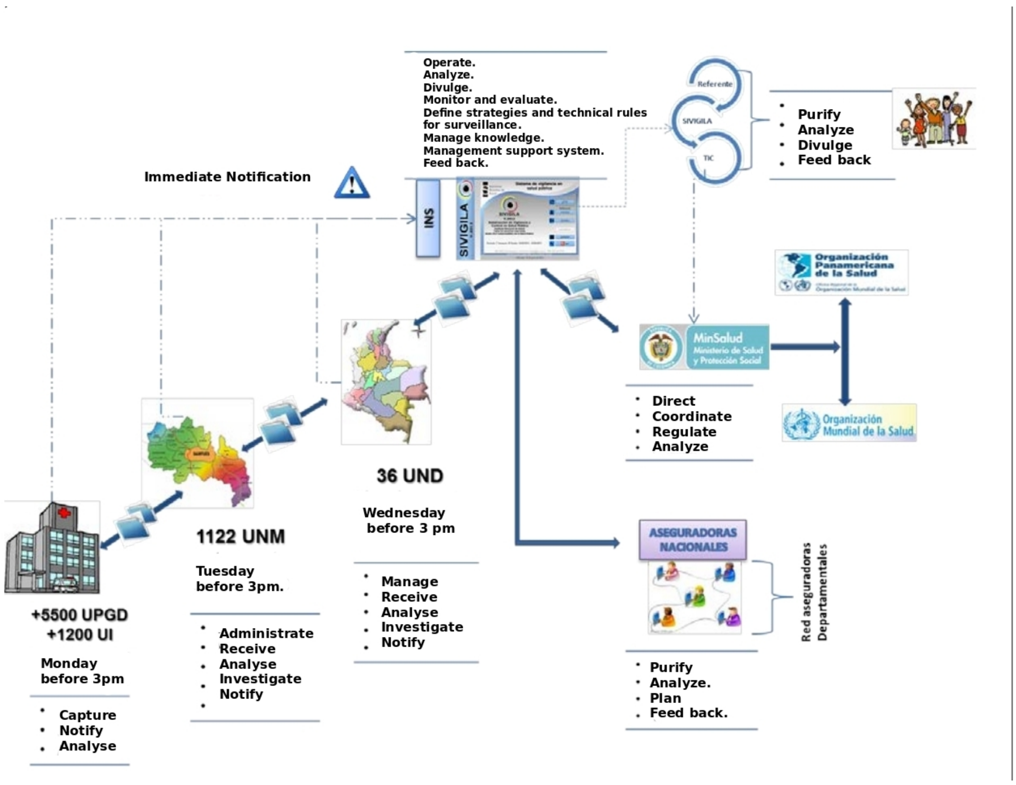


Figure 4.4: SIVIGILA Information Flow (Grupo Sistemas De Información en Salud Pública SIVIGILA 2014, own translation).

Activity	UI	UPGD	UNM	UND	UN
Install and update SIVIGILA	yes	yes	yes	yes	yes
Configure SIVIGILA	yes	yes	yes	yes	yes
Migrate information from previous years	yes	yes	no	no	yes
Determine UPGD	yes	yes	yes	yes	no
Register information	yes	yes	no	no	no
Notify	yes	yes	yes	yes	yes
Consolidate	no	no	yes	yes	yes
Realize adjustments	yes	yes	no	no	no
Generate Reports	yes	yes	yes	yes	yes
Feed back	no	no	yes	yes	yes

Table 4.3: SIVIGILA Units and tasks (Grupo Sistemas De Información en Salud Pública SIVIGILA 2014, own translation).

According to an engineer that works with SIVIGILA in the SDSC an example use case would be: A doctor fills out a form on paper or the SIVIGILA client (or both) for a given event of epidemiological interest. Then the form is saved in the application and is classified as immediate or normal notification. The information is then transferred to the municipal secretariat where it is validated and furthered transferred to the Departmental Secretariat for verification on if it is complete. It is then validated and consolidated and transferred to the INS. (Personal Correspondence, 22.3.2013)

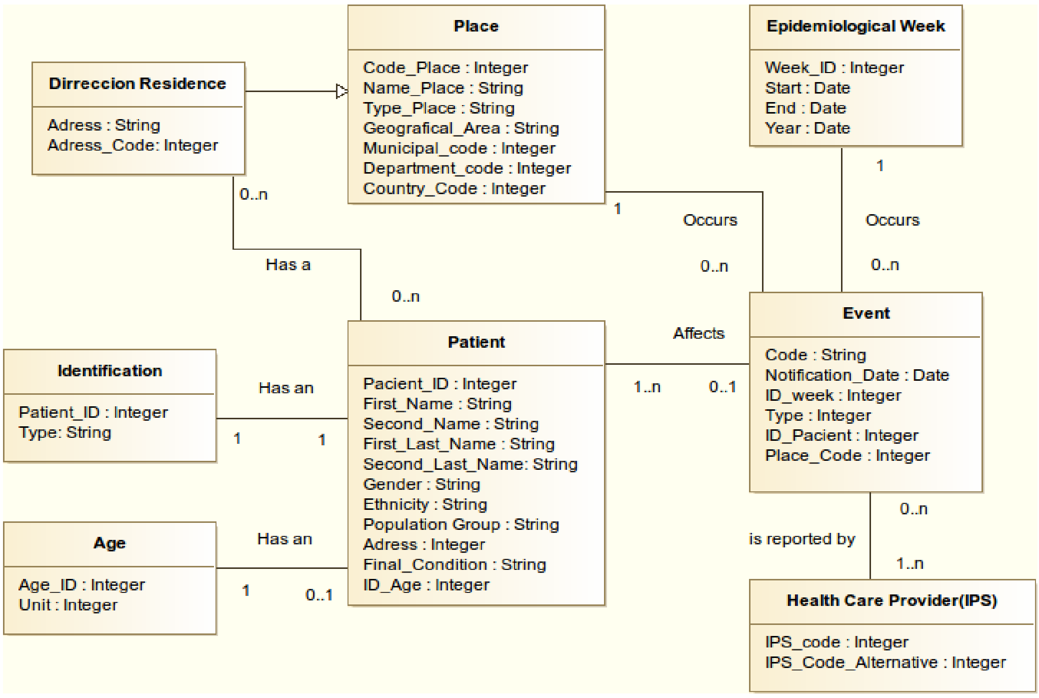


FIGURE 4.5: SIVIGILA Class Diagram. Adapted from (Bolaños & Muñoz 2009, own translation).

Figure 4.5 shows an UML class diagram for SIVIGILA the software[1]. Its main entities are Person, Public health Event and Health Service Provider. An Event affects a patient and occurs at a given Place and time. The event is reported by a Health Care Provider.

4.4.2 DHIS2

The District Health Management Software (DHIS2) was introduced to the secretariat of Health as a pilot in 2012. It is a DHIS 2 "is a tool for collection, validation, analysis, and presentation of aggregate and transactional data, tailored (but not limited) to integrated health information management activities." (*DHIS2* 2014). The software is free, open source and is developed by the HISP programme. It can be used as a a data repository for integrated health information systems that is described in chapter 2.

[1]See appendix E for a fuller model.

4.4.3 Other Health Information Systems

DANE, The National Administrative Department of Statistics is responsible for planning, collection, processing, analysis, and dissemination of Colombia's official statistics. DANE is a part of the executive branch of Colombian government and has research projects in economics, industry, population, agriculture and quality of life. (*About DANE* 2013).

The **Individual Record of Provision of Health Services(RIPS)** is the set of minimum and basic data for the General Social Security System (SGSSS) that is required for process management, regulation and control. The data from these registers relates to the identification of the health service provider (IPS). One of the Register's purpose is to facilitate trade relations between management entities (payers) and health provider institutions by presenting the details of the sales invoice with a structure that, being standard facilitates communication, data transfer processes and account reviews, regardless of the IT system used by each provider. RIPS can be used to formulate health politics, evaluate coverage and control health spending. (*Sistema de Informacioón de Prestaciones de Sauld - RIPS* 2013)

SISMED, the Information system of drug prices, is used to support of the policy of drug price regulation. It provides data that is necessary to analyze and control the drug prices in Colombia. (Suárez 2012)

The Registry of Affiliates **RUAF** is an information system that contains the record of the members that are affiliated to the Social Security System (health, pension, occupational hazards), family subsidy and beneficiaries of the programs provided through the social safety net, such as those offered by the SENA, ICBF, Social Action and other entities. (*Cuadro de temas Registro Único de Afiliados - RUAF* 2013)

Integrated Information System of Social Protection **(SISPRO)** is a system that collects information from various sources. SISPRO integrates information from the institutions that generates information for policy decision making, regulatory monitoring and managing services, with data warehouse technology. (*Sistema Integral de Información de la Protección Social - SISPRO* 2013)

In addition to these systems, there are several **health programme specific information systems**.

4.5 Interoperability Project

This section presents project background, collaboration, requirements and scope for my research.

4.5.1 Background

DHIS2 was implemented as a pilot in September 2012 in SDSC. According to an IT worker at SDSC, the SDSC organizational layout incorporates areas. Figure 4.6[2] shows the different areas in SDCD. The areas are responsible for one or more health programs. The programs have officers ('referentes' in spanish) and field workers at the municipalities. The officers inspect, supervise, control and do analysis, but also ensure that the institutions take the necessary actions to mitigate risks. (personal correspondence, 2014)

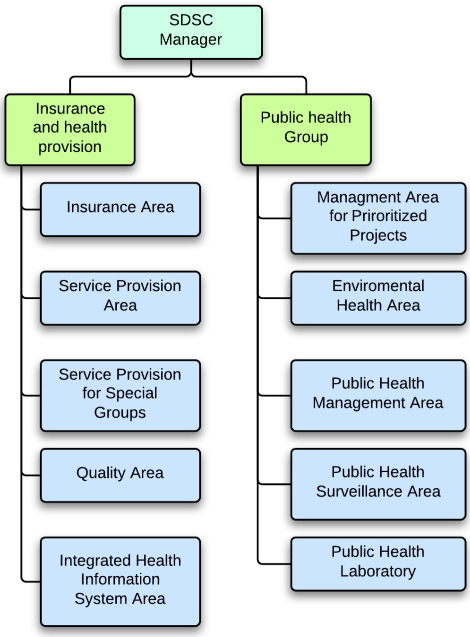


FIGURE 4.6: SDSC Areas.

An IT team from the secretariat of health in Cauca worked on the implementation of the pilot. The project began as a pilot implementation for a few health programmes. It was executed jointly between the University of Cauca and the Secretariat of Health in Cauca(SDSC). Some of the goals of the project were to: Adapt to the needs collection/reporting of health programs. Develop a map of health facilities to be integrated

[2]The figure is adapted (own translation) from a figure in SDSC webpage: saludcauca.gov.co. However, the webpage is down at the time of finalizing the thesis.

into a GIS. Improve data quality in terms of eliminating redundancies and duplication and by the use of validation. Improve data analysis of information by incorporating indicators from health programs. Enhance information visualization tools for decision makers in public health. (*Buena práctica: Sistema de Información de Salud con DHIS2 en Cauca* 2013, own translation)

Cauca's Departmental Secretariat of Health (SDSC) began using DHIS2 after considering other software packages to fit their needs. The important factors for choosing the software package were: Georeference of events, management of reports and indicators, web based user interface, interoperability, cost and implementation time (Cobo & Garcia 2012). The systems that where taken into consideration were:

- Google Maps
- Crowdmap
- Sigepi
- Tableau
- DHIS2

While Google maps is free, a negative aspect was time in development for the software. Sigepi had drawbacks when it came to interoperability with other systems. Crowdmaps would provide free usage with export and import of data. However it would be difficult to integrate with other systems. While Tableu satisfied most of the factors, it required a commercial licence. DHIS2 scored highest with free usage, web access, geographic location, import and export of data and interoperability with other systems. The IT-team in Cauca argued that the implementation of DHIS2 required experience in certain technologies but that support they would get from the University of Oslo. (Cobo & Garcia 2012)

Cauca's Departmental Secretariat of Health (SDSC) used two health programs in the pilot. The health programmes chosen were vector-borne diseases (ETV) and Sexual and reproductive health (SSR) in the municipalities of Inza, Tierra Adentro and Santander de Quilichao.

In January 2013 SDSC used SQL scripts, to transfer data from SIVIGILA to DHIS2. An IT worker in the SDSC adapted the code to fit the challenges at hand. The code was split in four files and was ran every-time they imported data from SIVIGILA to DHIS2, on a weekly basis. While it was good for the pilot there was a need for a stable solution. Interoperability between the two systems could improve the usage of DHIS2.

It could transfer the needed information in a standardized way. The maintenance of the code was complex and was also a motivating factor for improving the solution. On this background the SDSC decided to start an interoperability project.

4.5.2 Interoperability Project Motivation and Collaboration

The interoperability project is an IT project having SIVIGILA and DHIS2 as target systems. According to an IT worker at the SDSC, it is based on the fact that the SDSC must provide a solution that integrates information exchange between 30 applications. In addition the SDSC must provide data quality improvements, track the delivery of health services and ensure mobility of clinical records between institutions to improve service delivery. In that vein, the secretariat should involve not only institutional software, but also the medical record software in use at different centers in Cauca. (Personal correspondence 02.05.2014)

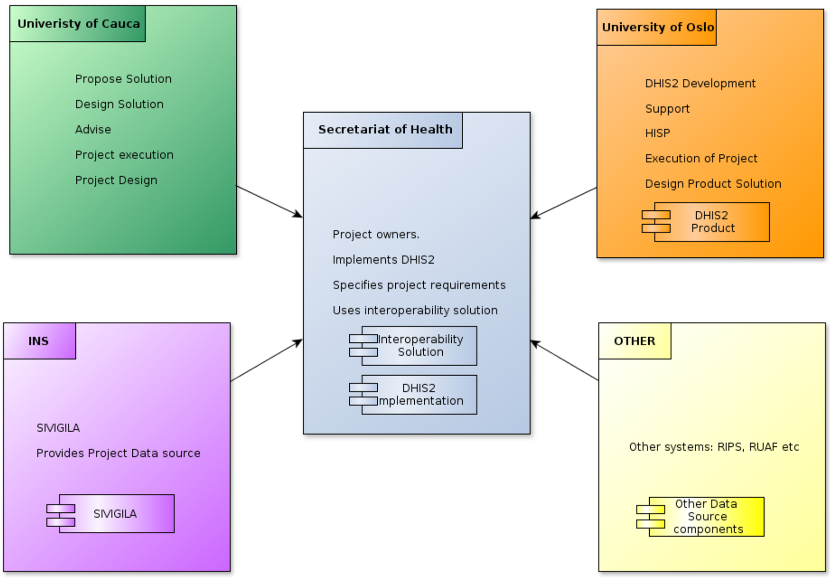


FIGURE 4.7: Involved Organizations and Source Components.

Figure 4.7 shows the collaboration for the interoperability project. The University of Oslo (UiO) provided design of the interoperability solutions, and execution of the project by me. UiO also provided indirect participation through support and DHIS2 product. The University of Cauca provided the main proposal and design for the interoperability solution. Project execution was provided by a University of Cauca master student that collaborated with me. His master thesis would continue our project. Advise and solution design was given by a Professor at the University of Cauca with research experience using HL7 standards. SIVIGILA provided the project data source. Finally the SDCD were the project owners. They implemented DHIS2, specified project requirements and were to use the interoperability solution. Other systems would be added later but our proof of concept only covers DHIS2 and SIVIGILA. SDSC specified requirements for the project. The next subsection presents the requirements.

4.5.3 Project Requirements

The following were the main requirements for the interoperability project:

1. A solution for interoperability between heterogeneous systems.
   (a) Establish a mechanism for optimal interoperability between SIVIGILA and DHIS2.
   (b) Improve the way data is transferred to DHIS2 from SIVIGILA.
   (c) Establish a solution that enables other systems to interoperate with DHIS2.
   (d) Establish a solution that can facilitate the portability of electronic health records in Cauca.
2. Georeference events.
3. Individual tracking.
4. A tool to support decision making by creating reports and by the use of indicators
5. A tool to support decision making in terms of analysis.
6. Congruity between health programmes.

Section 4.5.5 on the following page shows the scope of the project in terms of the requirements.

4.5.4 Tracking Individual Health Events

According to one of the members of the SCSD IT team, DHIS2 in Cauca required individual data due to the following reasons (Personal Correspondence, 2014):

- The software SIVIGILA requires a thick client installed at every work station and thus lacks a web solution for reviews of public heath events.
- Consultation of individual records is needed by supervisors in the area of epidemiological surveillance.

- In DHIS2, the forms for investigation and monitoring of events were created, for record tracking. For example: SIVIGILA reports a case of severe dengue and a review of the follow up form and investigation results that there was not dengue but malaria.
- There are some programs, that have several stages and with whom they could follow up on treatments using individual data.

4.5.5 Scope of the interoperability project

The research project scope covers requirement 1a,1b, 1c, 1d, and partly 1e. Requirements two through six are out of scope but play a peripheral role due the fact that requirement 1 would enable them.

The scope of the interoperability project is interoperability for DHIS2 and SIVIGILA with the use of HL7 CDA as a CDM. It incorporates sending data from SIVIGILA to DHIS2 without a return message (Fire and Forget)[3]. It mainly assumes a 'perfect scenario' and does not take into account errors from SIVIGLA.

There are some things that were out of scope such as making CDA implementation guides and templates. Castrillón et al. (2012) argue that a formal implementation guide should be done by a group of experts in clinical records and experts in CDA. Additionally, security, deployment and operation in general of the solution are also out of scope. The use DHIS2 to fit SDSC needs, the requirement of decision making, analysis and congruity between health programs entails health policy and resource management focus and is outside the scope of my work.

4.6 Summary

This chapter presented the setting for the interoperability project. It shows information on Colombia and Cauca. It described the health system of Colombia and various Health Information Systems. The history, collaboration, motivation, requirements and scope for the interoperability project were also presented. The information shown is the basis for two AR cycles that resulted in two interoperability implementations. The first is shown in chapter 5 on page 67 and the second in chapter 6 on page 79.

[3]For an explanation of fire and forget see Josuttis (2007, p.125).

Chapter 5

First Implementation

This chapter presents the First Implementation. It was part of the fieldwork done in Cauca. The collaboration efforts are shown in section 5.1. The proposed solution is presented in section 5.2. Different tools and a technical overview of the First Implementation are presented in section 5.3. The implementation in Cauca at the SDSC is described in section 5.4. Finally section 5.5 shows what happened after the implementation.

5.1 Collaboration

The execution of the First Implementation was done by me and another student/worker at the University of Cauca/SDCD. He was going to continue the work in his master thesis. We were responsible for setting up the project. Thus there was a collaboration between the University of Oslo, the University of Cauca and the SDCD as shows in figure 4.7 on page 63.

SDCD provided knowledge on SIVIGILA and access to relevant data. The University of Cauca provided the solution design, access to the HL7 CDA standard and experience with MIRTH Connect. I represented the University of Oslo in this part of the project. It was planned that other HIS would be included in the project at a later time.

I worked with creating the CDA instance. The other student/worker provided knowledge about flat files from SIVIGILA. We got one hour guidance sessions from Universidad de Cauca were we could discuss our advances and get advice. The advices had limmited effect on our practical work. This implementation included creating a HL7 CDA example instance on the basis of a SIVIGILA data form called "basic data" and using Mirth Connect.

5.2 Proposed Solution

The solution was proposed by the University of Cauca. The SDCD IT team, where the student/worker was a member, and me got the task to use a Hub and Spoke approach with MIRTH Connect. The solution included HL7 Clinical Document Architecture CDA example instance as a CDM.

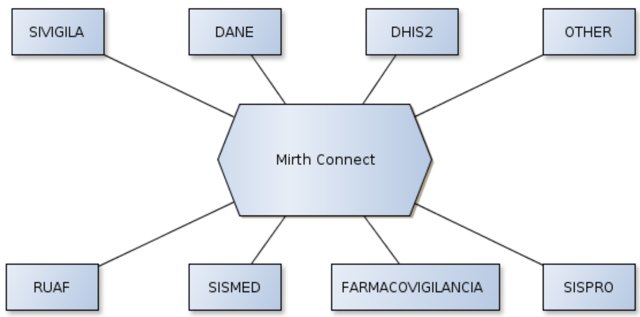


FIGURE 5.1: Vision of Hub and Spoke for SDSC.

Figure 5.1 shows the vision of the interoperability solution for the various systems.

The proposed design would cover requirement 1 a, b, c, d:

1. MIRTH Connect would be an Hub and Spoke for the SDCD heterogeneous systems. The use of the HL7 Clinical Document Architecture standard in combination with MIRTH could reduce the amount of interfaces required for every connection.

2. The solution would improve the way data is transferred to DHIS2 from SIVIGILA by offering loose coupling. The transfer could be done automatically rather than manually.

3. The solution would enable the mobility of electronic health records across the department of Cauca if the systems were added using the CDA as a CDM.

5.2.1 Implementation Scope

This implementation was intened as a proof of concept. The scope of the proof of concept covers SIVIGILA and DHIS2. The solution focused on the basic data form for SIVIGILA that is shown in figure 5.5 on page 74. There are 52 other data forms for various events that would be relevant if the solution were to be deployed and expanded to cover all SIVIGILA event types.

5.3 Implementation Tools

Several tools were used to implement the interoperability solution such as Mirth Connect, a database tool and a tool used for developing a CDA instance.

5.3.1 Mirth Connect

As stated earlier we used MIRTH Connect, an open source "Healthcare integration engine, specifically designed for HL7 message integration" (*MIRTH* n.d.). It is designed for "exchange of patient-specific information, such as laboratory results and medical records" and employs a HL7 messaging protocol.

Mirth Connect uses an interface model that is composed of a Channel that contains a Source Connector, Filter, Transformer and Destination Connectors (Bortis 2008). The Mirth Channel is shown in figure 5.2 with the said components. Figure 5.3 shows different ways Mirth can be used such as broadcast, router and chaining in addition to the basic integration mode shown in figure 5.2 on the following page [1]

The Source connector connects to a source system and receives data. Filters can filter out messages based on rules. The Validation component checks the validity of the message structure. Transformers can transform data from one format to another. Destination

[1] See also "Mirth Webinar" for an explanation of the figures, http://www.mirthcorp.com/webinars/mirth-connect-webinar-part-1-mirth-connect-overview

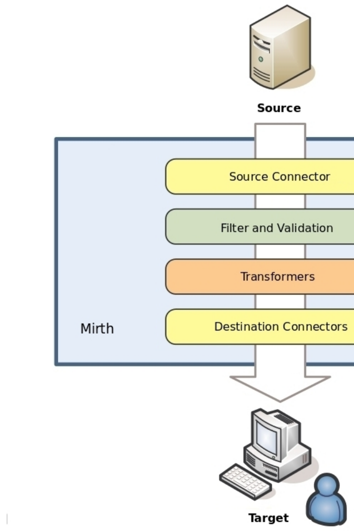


FIGURE 5.2: Mirth Connect Channel Overview. Adapted from (Bolaños & Muñoz 2009, own translation).

connectors can be one or many for a given channel. They connect to the target system and can insert values. (Bortis 2008)

While Mirth Connect is free to download, training courses cost 3000$ per person (3 days in California). Access to online tutorials is open only for supporting customers. MIRTH support costs between 5000$ to 30 000$ annually.[2]

There is a wiki and some videos (webinars) that one can use for free but they are not detailed. Additionally, there is a CDA API library in Mirth Connect but it requires a commercial licence.

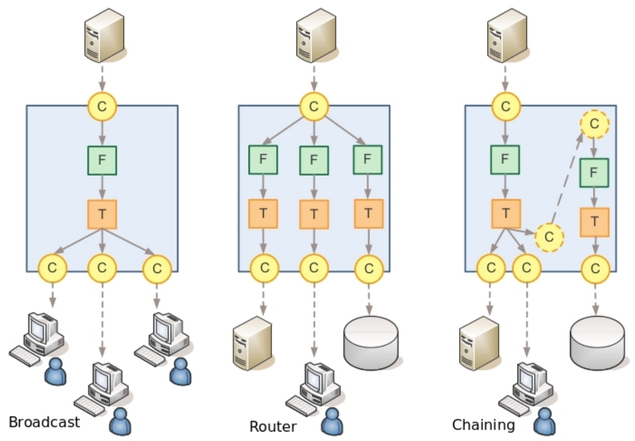


FIGURE 5.3: Mirth Connect channel modes (Bolaños & Muñoz 2009, own translation).

[2] For a detailed overview of Mirth Connect pricing see: http://www.mirthcorp.com/products/mirth-connect#Pricing

Mirth Connect supports different message standards and transfer protocols as shown in table 5.1 and 5.2. The relevant message standards for the implementation is HL7 v3. The relevant transfer protocols are Files and Database.

Message Standards
HL7 v2.x
HL7 v3
XML
X12
EDI
DICOM
NCPDP
Delimited Text

Table 5.1: Supported Message Standards.

Transfers Protocols
MLLP
TCP/IP
HTTP
Files
Database
S/FTP
Email
JMS
Web Services
PDF/RTF Documents
Custom Java and JavaScript

Table 5.2: Supported Transfer Protocols.

5.3.2 CDA Eclipse plugin

An Eclipse plugin called Eclipse Instance Editor was used for the development of a CDA instance. It can be "be used to both edit, as well as validate, CDA instances" and "the validation process is based on the MIF of the CDA R-MIM" (*Eclipse Instance Editor* 2009). The tool gives explanations for errors. It also has code highlighting for CDA in the official HL7 RIM colours and a design tab with a CDA field table.

5.3.3 Database Tool

Importing the data from a CSV file to a Postgres database was performed with a specialized tool. A team member did work with this tool, I couldn't record the details.

5.4 Interoperability Implementation

The work on interoperability began on January 2013 upon my arrival in Cauca. Mirth Connect was chosen because Universidad del Cauca had experience with it in previous research projects using HL7 v.3.

The time line of the solution had four parts. We had one week to setup the source and destination connectors and another to create the CDA instance. Upon its completion,

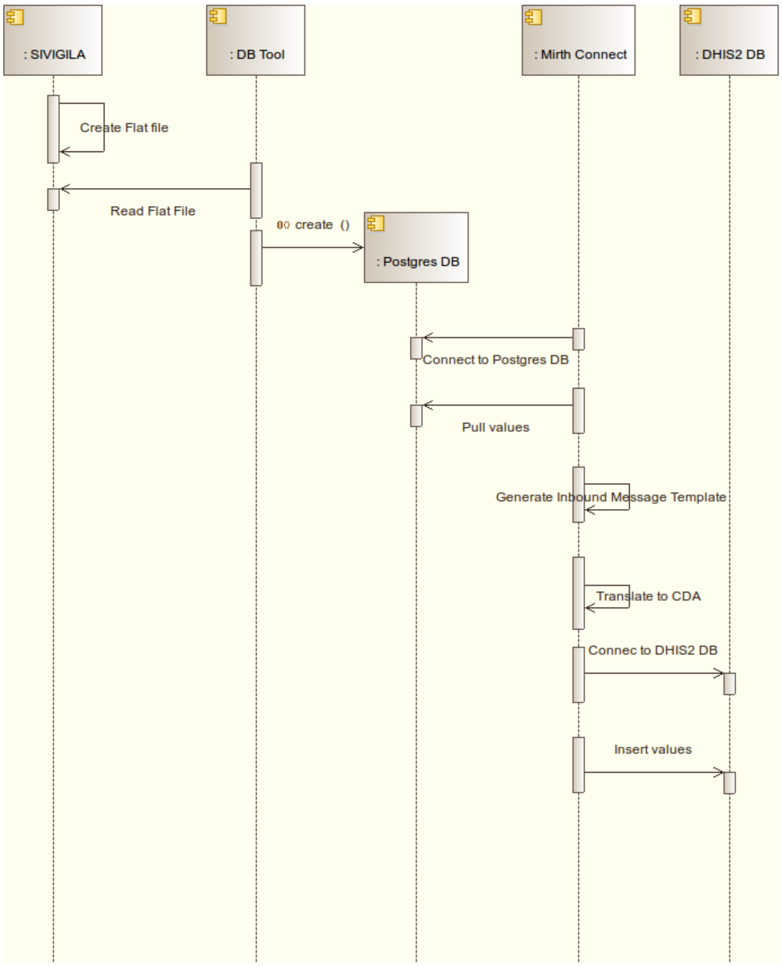


FIGURE 5.4: Sequence Diagram for the solution.

the plan was to start mapping the values from SIVIGILA to the CDA and finally write to the DHIS2 database. There was also some preliminary propositions to write a CDA template and an implementation guide but due to complications it was not prioritized.

The solution can be seen in figure 5.4 with SIVIGILA, a database tool, a temporary Postgres database, Mirth Connect and the DHIS2 database. SIVIGILA creates flat files weekly. The files contain information about health events. They are described in section 5.4.1. A Database tool was used by a team member to create a Postgres database storing the information from the flat files containing patient and organisation unit data. Thus making it possible for Mirth Connect to connect to the database, pull values from the database and automatically generate an inbound XML message based on the selected values from the database. The message gets translated to a CDA instance. Finally the destination connector connected and insert values to the DHIS2 database.

5.4.1 The Flat files

The flat files[3] that SIVIGILA produced contained data that should be imported into DHIS2 on a weekly basis. The ones we had available for the proof of concept contained:

- UPGD data (used in the proof of concept).
- Data on the human talent available for monitoring and service delivery in primary units generating the data UPGD.
- Individual weekly notifications with patient data (used in the proof of concept).
- Weekly collective notifications.
- Control data.
- Data on events that are notified individually.
- Collective morbidity events.

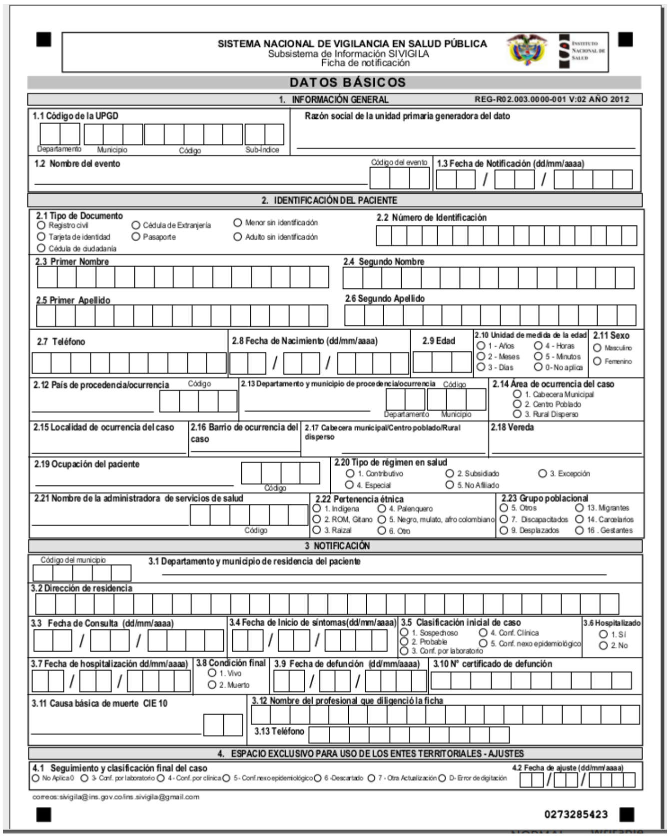
Two files were relevant for the solution. They contained data about the UPGDs and individual patient data. There was no description of the different fields in the flat files. The field names were found with help of another worker in SDSC. We were only able to find names for relevant fields. They contained data on patient demographics with a SIVIGILA event and a possible cause of death that was coded with ICD-10. The values would later be used in a CDA instance.

5.4.2 Development of the CDA Instance

A Mirth Outbound Message Template was developed using the CDA schema, CDA HMD and on the basis of the SIVIGILA data form 'Basic Data'. Figure 5.5 shows the data form. I had access to a used form but it contains sensitive data and is not shown. I discovered that the SIVIGILA form was from 2009 after finishing the CDA instance. Thus, the CDA instance had to be adapted to the newer 2012 version.

I based my work on a CDA tutorial, the CDA documentation, example CDAs from other projects and the CDA XML schema. The CDA XML schema (XSD) was used in Eclipse to see the element relationships, different components and other information, for example if an element or attribute is required. I had to combine the use of the CDA HMD and the XSD because they had different strengths and weaknesses. The former gave a quick overview but had less detail than the latter.

[3]See Grupo Sistemas De Información en Salud Pública SIVIGILA (2014, p.9) for details on the files.


SISTEMA NACIONAL DE VIGILANCIA EN SALUD PÚBLICA
Subsistema de Información SIVIGILA
Ficha de notificación

DATOS BÁSICOS

1. INFORMACIÓN GENERAL — REG-R02.003.0000-001 V:02 AÑO 2012

1.1 Código de la UPGD (Departamento, Municipio, Código, Sub-Índice) — Razón social de la unidad primaria generadora del dato

1.2 Nombre del evento — Código del evento — 1.3 Fecha de Notificación (dd/mm/aaaa)

2. IDENTIFICACIÓN DEL PACIENTE

2.1 Tipo de Documento: Registro civil; Tarjeta de identidad; Cédula de ciudadanía; Cédula de Extranjería; Pasaporte; Menor sin identificación; Adulto sin identificación — 2.2 Número de Identificación

2.3 Primer Nombre — 2.4 Segundo Nombre

2.5 Primer Apellido — 2.6 Segundo Apellido

2.7 Teléfono — 2.8 Fecha de Nacimiento (dd/mm/aaaa) — 2.9 Edad — 2.10 Unidad de medida de la edad: 1 - Años; 2 - Meses; 3 - Días; 4 - Horas; 5 - Minutos; 0 - No aplica — 2.11 Sexo: Masculino; Femenino

2.12 País de procedencia/ocurrencia (Código) — 2.13 Departamento y municipio de procedencia/ocurrencia (Código: Departamento, Municipio) — 2.14 Área de ocurrencia del caso: 1. Cabecera Municipal; 2. Centro Poblado; 3. Rural Disperso

2.15 Localidad de ocurrencia del caso — 2.16 Barrio de ocurrencia del caso — 2.17 Cabecera municipal/Centro poblado/Rural disperso — 2.18 Vereda

2.19 Ocupación del paciente (Código) — 2.20 Tipo de régimen en salud: 1. Contributivo; 2. Subsidiado; 3. Excepción; 4. Especial; 5. No Afiliado

2.21 Nombre de la administradora de servicios de salud (Código) — 2.22 Pertenencia étnica: 1. Indígena; 2. ROM, Gitano; 3. Raizal; 4. Palenquero; 5. Negro, mulato, afro colombiano; 6. Otro — 2.23 Grupo poblacional: 5. Otros; 7. Discapacitados; 9. Desplazados; 13. Migrantes; 14. Carcelarios; 16. Gestantes

3 NOTIFICACIÓN

Código del municipio — 3.1 Departamento y municipio de residencia del paciente

3.2 Dirección de residencia

3.3 Fecha de Consulta (dd/mm/aaaa) — 3.4 Fecha de Inicio de síntomas (dd/mm/aaaa) — 3.5 Clasificación inicial de caso: 1. Sospechoso; 2. Probable; 3. Conf. por laboratorio; 4. Conf. Clínica; 5. Conf. nexo epidemiológico — 3.6 Hospitalizado: 1. Sí; 2. No

3.7 Fecha de hospitalización (dd/mm/aaaa) — 3.8 Condición final: 1. Vivo; 2. Muerto — 3.9 Fecha de defunción (dd/mm/aaaa) — 3.10 N° certificado de defunción

3.11 Causa básica de muerte CIE 10 — 3.12 Nombre del profesional que diligenció la ficha — 3.13 Teléfono

4. ESPACIO EXCLUSIVO PARA USO DE LOS ENTES TERRITORIALES - AJUSTES

4.1 Seguimiento y clasificación final del caso: No Aplica 0; 3- Conf. por laboratorio; 4- Conf. por clínica; 5- Conf. nexo epidemiológico; 6 -Descartado; 7 - Otra Actualización; D- Error de digitación — 4.2 Fecha de ajuste (dd/mm/aaaa)

correos: sivigila@ins.gov.co/ins.sivigila@gmail.com

0273285423

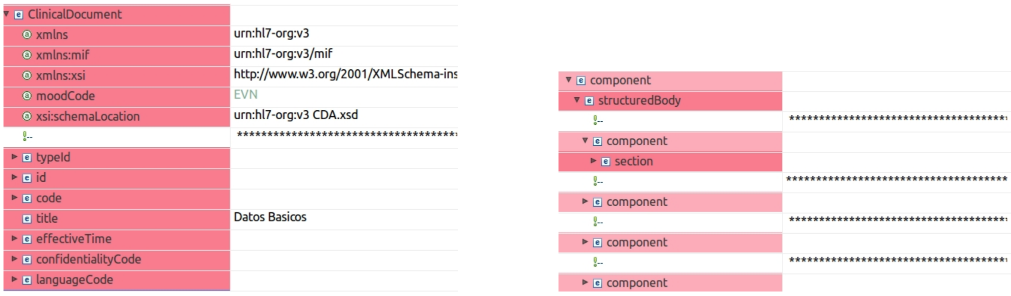
FIGURE 5.5: SIVIGILA Basic data Form.

Appendix A shows the example CDA that was developed. The CDA contained a header, body and participants. Figure 5.6(a) on the next page shows the elements in the header. Figure 5.6(b) on the following page shows the body choice: structured body. It allows XML-encoded data. Four sections were used in the body: 'General information', 'Patient Identification', 'Notification' and a section for information adjustments. These sections contained a list of items. The items were the fields in the SIVIGILA basic data form.

Figures 5.7(a) and 5.7(b) show the participants in the CDA instance: Record target (patient), author (medical staff), custodian and information recipient (SDSC). The figures are taken using the eclipse instance editor. There was no time for in depth learning on these topics.

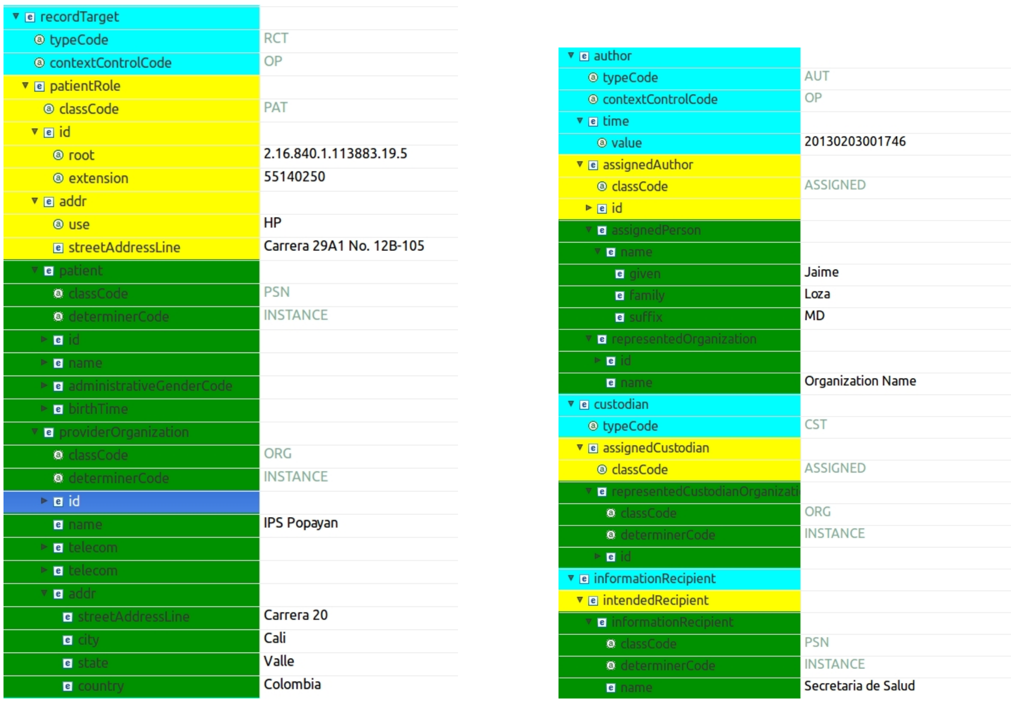
I found the following issues regarding codes: Colombia does not have any Object Identifier (OID) for the specific elements I needed in the CDA. However Colombia does have an OID code for the country as a whole: 2.16.170[4]. Rather than making examples from the Colombian OID I chose to use a HL7 example OID. The only code set used was LOINC. There was no time to use ICD-10 or define own OID.

[4]See OID Repository entry for Colobmbia for details, http://www.oid-info.com/cgi-bin/display?oid=2.16.170&action=display



(a) CDA Header. (b) CDA Body.

FIGURE 5.6: CDA Header and Body with standard coloring.



(a) CDA Record Target. (b) CDA Author, Custodian and Information Recipient.

FIGURE 5.7: CDA Participants with standard coloring.

5.4.3 Setup and Work

The main work can be divided in phases for installing and setting up Mirth Connect, configuring the source connector, translating, and setting up the destination connector. MIRTH Connect was installed on a Linux machine and a co-worker installed it on Windows to develop the prototype. A PostgreSQL database was created for the Mirth Connect database.

An early attempt to connect Mirth Connect directly to SIVIGILA FoxPro database turned impossible because FoxPro is incompatible with Mirth Connect Source Connector. Thus the usage of flat files described in section 5.4.1 on page 73.

Mirth Connect's destination connector gave errors when data was inserted into the DHIS2 database. The errors were found when trying to insert several values at a time rather than just one. Mirth Connect read several lines instead of one.

Another problem was that the flat files contained wrong UPGD identifiers. The SDSC IT team wanted to allow the transfer the events independently if there were erroneous organization units identifiers. Therefore two flat files were needed, one for patient data and another for UPGD. However Mirth Connect can only read one input file per Source Connector and supports just one Source Connector per Channel. We solved this issue by using the database tool to merge the two files thus creating a Postgres database. The Source Connector was configured and connected to read this database. The values were automatically mapped to Mirth Connect inbound XML message. The selected values were patient names, gender and event.

The translation was done by mapping the values from the inbound message to a CDA instance that was used as an outbound message.

The values in the CDA could thus be used in the Destination Connector.

However, we managed to insert one patient only. An error occurred when inserting values due to duplicate patient identifiers. To generate different identifiers we tried to use java script in Mirth Connect but it proved unsuccessful. The DHIS2 API was not considered because we would reuse SQL code from a previous attempt as explained in subsection 4.5.1 on page 61.

This First Implementation ended up with a solution that was far away from the one planned and could not fulfil the requirements.

5.5 Post Mirth

After the failure of the First Implementation the SDSC IT team began using a different software called Talend to transfer data from SIVIGILA to DHIS2 without a CDM. This work started after my stay in Cauca. MIRTH Connect was not useful because of the lack of free extensive documentation, its price, the complexity of the CDA and technical problems when writing directly to the DHIS2 database.

5.6 Summary

This chapter presented a collaboration process, a solution and the tools used for the First Implementation. Mirth Connect was used in combination with an eclipse plugin and a database tool to create the solution. It was so difficult to use Mirth Connect to make the target systems interoperable that the solution was discontinued. The next chapter presents a Second Implementation based this experience.

Chapter 6

Second Implementation

This chapter describes the second interoperability implementation. Section 6.1 describes a different approach and its collaboration process. Section 6.2 presents the solution design using Hohpe & Woolf (2003) patterns. Section 6.3 shows different Mule Concepts that are relevant for the implementation. Section 6.4 describes the developed solution, the implementation of the design. Section 6.5 presents details on the adapters in the ESB implementation. Section 6.6 shows how I used the CDA as a CDM and the challenges I encountered. Section 6.7 shows problems I met when developing the solution. Section 6.8 describes the CDM conversion.

6.1 A Different Approach

After the failure of the First Implementation with Mirth Connect, I carried out a a second implementation with a different architectural approach.

One limitation of the project that had to be taken into account was the limited amount of funds. The cost of HL7 integration interface engines such as Mirth Connect and Iguana would surpass the available funds. The HL7 modules of alternative integration software are usually also expensive. This fact in combination with the complexity of the CDA standard meant that it took a while before I realized that it was possible to carry out the project with limited funds.

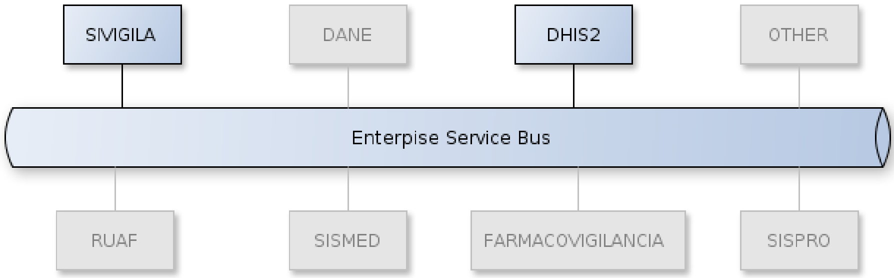


FIGURE 6.1: Architectural Approach for the Second Implantation.

Figure 6.1 shows the architectural approach for the second implementation. It is composed of SIVIGILA and DHIS2 with an Enterprise Service Bus. The other systems would also use the ESB as an interoperability solution rather than than a Hub and Spoke (Mirth Connect). Collaboration was crucial for developing the second implementation. The following subsection describes the collaboration.

6.1.1 Collaboration

The second implementation was done in Norway and the collaboration was limited to communication with the IT team in SDSC via emails, chat and Skype calls. This type of collaboration lead to some confusion and waste of time mainly due to synchronicity problems and the time it took to solve transmission problems. An example of the latter was a backup copy of the DHIS2 database that was sent twice due to a corrupt file in the first transmission. Discarding other error sources and waiting for a correct copy resulted in waist of time. The synchronicity problems arose mainly in the time difference between Colombia and Norway and the different tasks performed by different members of the SDSC IT team.

Later, when the Second Implementation was working, I gave them support and provided examples on how to use DHIS2 web API to alleviate the amount of work they were facing. At that time they were still using Talend. Their Talend jobs inserted data directly to the DHIS2 database rather than using the API. One of the SDSC engineers mentioned that the Talend solution required much maintenance: A new DHIS2 version required a lot of re work.

6.2 Initial Interoperability Solution Design

I created a solution design using Hohpe & Woolf (2003) patterns. It entailed the use of CDA as a canonical data model and translators as shown in to the figure 6.2. The design focused on how the different components would work rather than if it would be a Hub and Spoke or Bus Architectural Approach. The source systems would periodically provide data files in their own formats. The CDA based canonical model would have a homogenizing function so that only adapter would push the information into DHIS2 regardless of the source system.

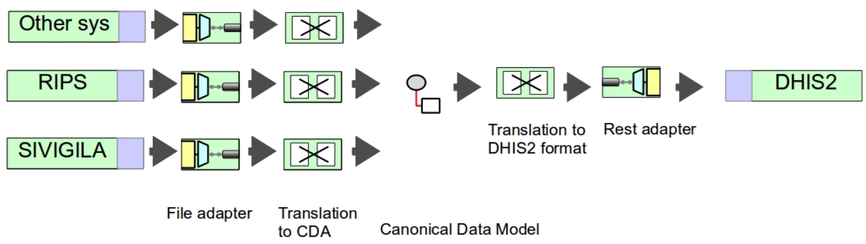


FIGURE 6.2: Interoperability Design for Many Systems.

Figure 6.2 shows the design for having many systems interoperating with DHIS2 with a CDA schema as a CDM. There was some uncertainty on which systems there would be because it requires knowledge on what data they need to transfer. However after communication with the SDSC team I found out that they needed individual data from RIPS and RUAF. These systems are candidates for CDA usage.

Figure 6.3 on the next page shows the planned design for the interoperability solution. A file adapter would be used to read the a SIVIGILA flat file. The file would be translated line by line so that each line would become a Clinical Document in CDA format. The CDA would be used as a CDM which then would be translated to DHIS2 format. Finally a rest adapter would POST the data to DHIS2 using its REST services. An interoperability software supporting this functionality was clearly needed to implement the solution, this meant an Enterprise Service Bus.

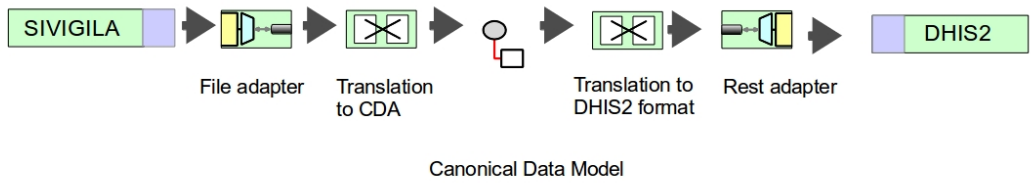


FIGURE 6.3: SIVIGILA and DHIS2 Interoperability solution Design with Scope.

6.2.1 Finding the right Enterprise Service Bus

The main challenge for choosing the right ESB was its support for the CDA or its XML schema (or XSD). As mentioned in section 6.6 on page 91, this XSD has technical flaws preventing its use in a translator.

I searched for an interoperability software that would work with CDA. It had to be free and easy to use and have extensive available documentation. Mirth Connect was excluded because it was too complex, costly, had problems in the First Implementation and was unable to satisfy the requirements. The use of a CDA API in MIRTH requires 30.0000$ a year according to a Mirth Account Executive (personal communication, 18.6.2014).

The working interoperability solution in Cauca used Talend (*Talend* n.d.) at the time, making it my first choice. However its HL7 components required paying for a different version of Talend. Talend had available extensive documentation and was easier to use compared to Mirth Connect. A similar component to the HL7 component in a free edition is called 'tFileOutputMSXML'. Thus Talend was installed and configured for translating the SIVIGILA format to CDA. I imported the CDA XSD upon which Talend crashed when saving the solution. The editions of Talend I worked with were Talend ESB and Talend Open Studio for Data Integration.

This problem lead to the abandonment of Talend and I proceeded to install and test Fuse IDE (*Fuse* n.d.) but the documentation was not as extensive as Talends and it lacked a graphical mapper component. Thus it seemed like the amount of coding would be higher and Fuse IDE was disregarded.

Mule was the 3rd alternative. However, its data mapper also crashed when loading the CDA XSD. I found a workaround by binding the XSD to Java POJOS outside Mule. This binding became the foundation of the CDM from now on. The POJOS were loaded into the mapper in two attempts.

For the first attempt I had to manually change one element to the CDA XML schema, but changing the CDA XSD was unacceptable.

The second attempt included using JAXB bindings to avoid changing the XSD. It is described in section 6.6 on page 91. The data mapper crashed in the second attempt also. I decided that the solution would not use a graphical mapper and instead use custom java translators for the mapping.

Mule Studio offered a layout and flow concept that made it easier to work with. Additionally it had a seamless developer interface for working with Java code and its layouts and flows. And Mules documentation is free, extensive and easy to read. On that basis I chose Mule.

While Mule had many of the qualities I was looking for, it lacked a message bus. I needed to find one.

6.2.2 Finding a Message Bus

A message bus would enable the different adapters to send messages to each other. I found a Mule tutorial on the message bus ActiMQ and decided to use it. Installing and configuring ActiMQ in Mule as a Global element and the use of JMS endpoints components was stright forward.

ActiveMQ "is an open source, JMS 1.1 compliant, message-oriented middleware (MOM) from the Apache Software Foundation" (Snyder et al. 2009, p.39). I could choose to use a queue or a topic for ActivMQ. The chosen MOM domain was a queue because it is simpler to configure. However, one could use a topic in the future.

6.3 Mule Concepts

Mule began as a SourceForge project in April 2003. Version 1.0 was released in 2005 and version 2.0 with many changes some years late (Dossot & D'emic 2010). The concepts of Mule are derived from Enterprise Integration Patterns (Hohpe & Woolf 2003, Dossot & D'emic 2010). It is a "java-based enterprise service bus (ESB) and integration platform that allows developers to quickly and easily connect applications to exchange data following the service-oriented architecture (SOA) methodology" (*Meet Mule* n.d.).

Figure 6.4 on the following page shows an example of how Mule can be used with six systems. A messaging bus like JMS or AMQP can be used to decouple applications from each other. (*What is an ESB* n.d.). Mule offers the following interoperability principles:

- Orchestration

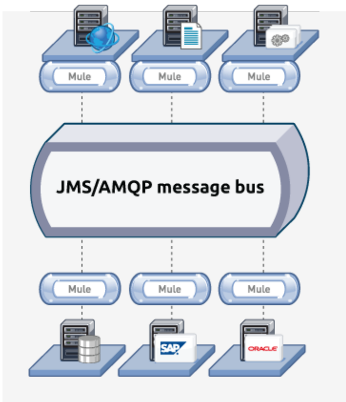


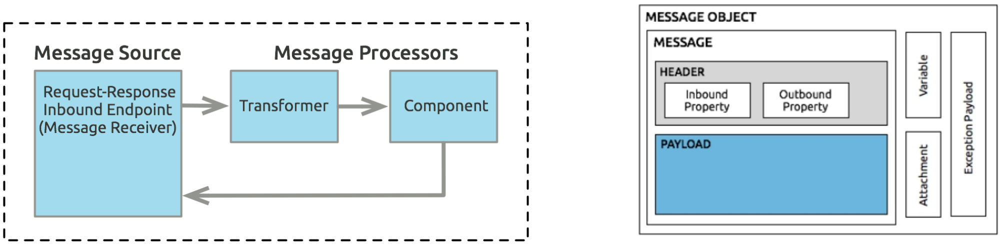
FIGURE 6.4: Mule Example ESB Implementation (*What is an ESB* n.d.).

- Transformation (translation)
- Transportation
- Mediation
- Non-functional consistency

Some of the relevant concepts that Mule uses are flows, elements and messages. According to Mules documentation a flow is "the construct within which you link together several individual elements to handle the receipt, processing, and eventual routing of a message" (*Mule Concepts* n.d.). A Mule flow is shown in figure 6.5(a) on the next page contains message processors(elements) that accept and process messages.

The elements in a mule flow can be message sources or message processors. A flow needs a message source that has a message from an external source. Message processors can be categorized as: connectors, components, transformers, (translators) filters, routers, scopes and exception strategies.

Another central concept to Mule is the Mule message. Its structure is shown in figure 6.5(b) on the following page. It has a message header with inbound and outbound properties, payload, attachment, variable and exception payload. The payload can be a serialized java object, XML strings or other. The header contains meta-data about the Mule message, the payload contains business-specific data. These concepts were used in the interoprability solution



(a) Mule Flow (*Mule Concepts* n.d.).

(b) Mule Message (*Mule Message Structure* n.d.).

FIGURE 6.5: Mule Message and Flow.

6.4 Interoperability Solution

The interoperability solution developed with Mule and ActiveMQ for SIVIGILA and DHIS2 is shown in figure 6.6 on the next page in a UML sequence diagram. It contains two mule adapters that were developed to connect to each system and translate messages.

SIVIGILA creates weekly a flat file described in the previous chapter. The Mule SIVIGILA Adapter reads the file and translates it from the SIVIGILA format to the CDA canonical data model. After the translation it sends the message to ActiveMQ as a JMS message. ActiveMQ transfers the JMS message from a JMS producer (SIVIGILA adapter) to the JMS consumer, that is the Mule DHIS2 adapter.

The Mule DHIS2 adapter translates the CDA java object that is received from ActiveMQ to three JSON messages: Person, Enroll and Event. They are finally sent to DHIS2 in three REST calls. The adapters are described in more detail in following section.

6.5 Adapters in Mule

The main function of two Mule adapters and the message bus is to create loose coupling. The adapters are responsible for translating to and from the CDA canonical data message. Between them, the message bus is responsible for transferring the message making each adapter independent of the other.

6.5.1 SIVIGILA Mule Adapter

The SIVIGILA Mule adapter was developed as a Mule flow. It is shown in figure 6.7 on the following page. It contains 6 elements: A file inbound-endpoint, A file to string transformer, a line splitter, a line to CDA translator, a logger and a JMS consumer outbound-endpoint.

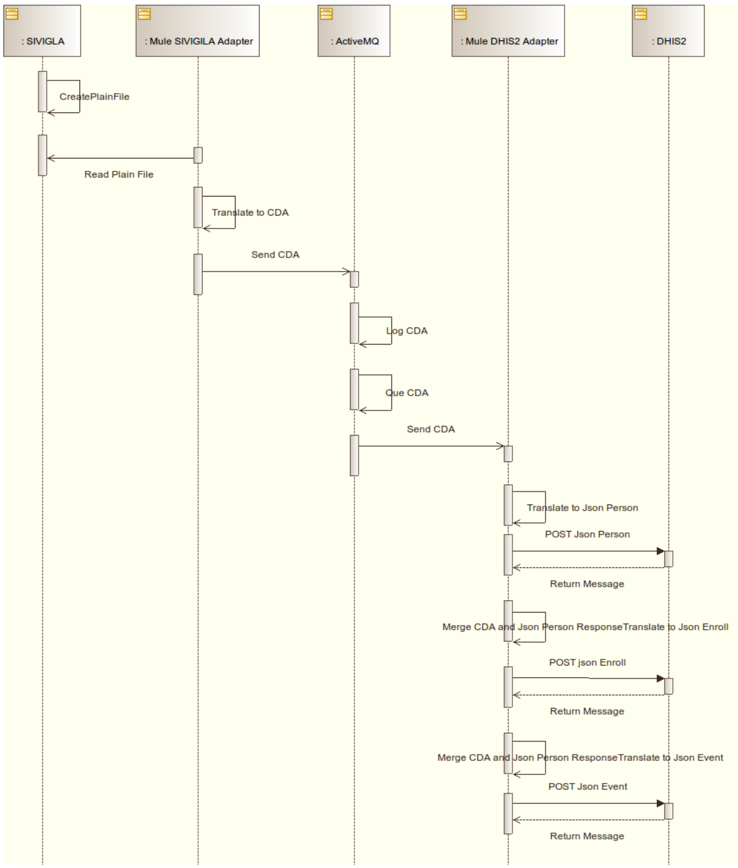


FIGURE 6.6: Simplified Interoperability Solution Sequence Diagram.

The SIVIGILA file inbound-endpoint reads the CSV files that SIVIGILA creates. It is similar to the one that was used in the first implementation. However there were an initial complication: It took time to grasp all the column names and their meaning.

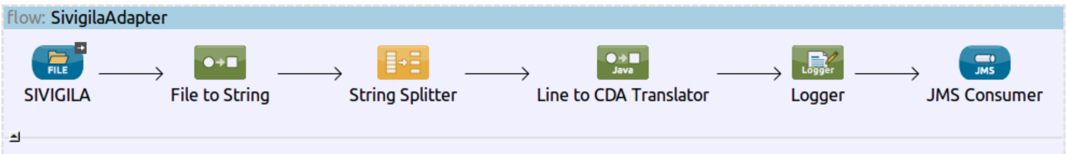


FIGURE 6.7: SIVIGILA Mule Adapter Flow.

As the collaboration with Cauca picked up and I got access to newer SIVIGILA files. The newer files came in two formats: an encrypted CSV file and a spredsheet from which I could extract the data I needed. According to a SDSC worker, the reason for encryption was that it was common to change its content.

The file inbound endpoint was configured to read the SIVIGILA CSV file. A Mule translator was used to transform from the file into String. The String Splitter splitted the string into single lines. It was done using the MULE script language MEL. Listing 6.1 shows the MEL code used in the element.

```
<splitter
  expression="#[rows=StringUtils.split(message.payload, '\n');↩
    ArrayUtils.subarray(rows,1,rows.size())]"
doc:name="String Splitter" />
```

Listing 6.1: Line Splitter in Mule Flow.

The Line to CDA Translator was a custom Java element written to map the values from a SIVIGILA CSV line to the CDA canonical data model. It processed each single line from the SIVIGILA file as one CDA message. The splitMessage() method in the translator was used to create a Hashmap containing the column name from the CSV file as keys and field values from the line as values. The values from the hashmap were then set in the CDA Java objects obtained from binding the the POCDMT000040-ClinicalDocument Java class.

Appendix B on page 135 presents the relevant derived java classes from the CDM. Appendix C on page 140 shows simplified sequence diagrams of how the custom translator creates the relevant objects and maps to the required classes. The diagrams show just the most relevant objects. The main objects that are injected into the clinicalDocument are POCDMT000040RecordTarget, POCDMT000040Author, POCDMT000040Custodian and POCDMT000040Component2. The translator returns a POCDMT000040 ClinicalDocument Java Object that is logged in the logger component and later sent to a JMS endpoint.

The JMS endpoint was defined by first creating an ActiveMQ global flow element in the Mule Project. Due to the fact that I was developing interoperability for two systems and the time constrain for this implementation, I chose a queue rather than a topic. The JMS endpoint was configured to use ActiveMQ. I accesed the ActiveMQs administration page to monitor the queue. Due to time constrains I decided to send Java objects to ActiveMQ rather than serialized XML.

This adapter implemented the input side of the whole process. The information it produced was channeled by ActiveMQ and transferred over to the output side in the DHIS2 Adapter.

6.5.2 The ActiveMQ Bus

The ActiveMQ component both separated and integrated the SIVIGILA and the DHIS2 adapters. It was introduced to prepare the interoperability implementation to include adapters for other systems, thus the separation. At the same time it integrated both adapters (and other to come) into a whole interoperability solution.

The ActiveMQ component both separated and integrated the SIVIGILA and the DHIS2 adapters. It was introduced to prepare the interoperability implementation to include adapters for other systems, thus the separation. At the same time it integrated both adapters (and other to come) into a whole interoperability solution.

6.5.3 The DHIS2 Mule Adapter

Appendix D shows simplified sequence diagrams of how custom java translators transfers information from the CDM. The DHIS2 Mule adapter needed to cover calls to three DHIS2 services. Its main challenge was to use the original input object through a branched flow. This mule flow became more complex. Due to its size it is split into three parts for practical description reasons.

6.5.3.1 Part one

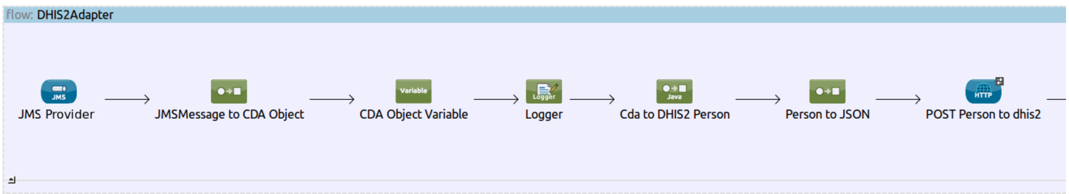


FIGURE 6.8: DHIS Mule Adapter Flow Part One.

Figure 6.8 shows the elements of the first part. The JMS Provider inbound endpoint was configured in a similar way as the JMS outbound endpoint in the SIVIGILA adapter. The JMS message to CDA Object translator translates the JMS message it receives to an POCDMT000040-ClinicalDocument Java object.

The CDA to Object Variable element receives the POCDMT000040-ClinicalDocument Java object in a Mule Message payload. The element sets a flow variable with the payload using Mule's MEL language as shown in Listing 6.2. There were alternatives to a flow variable, as using a session variable or a message property.

```
    <set-variable variableName="CDAVAR" value="#[message.payload]" ↩
  doc:name="Variable"/>
```

LISTING 6.2: Setting the Mule Flow Variable.

I chose to use a flow variable to be able to reuse the same object in all branches described in parts two and three. Listing 6.3 shows how I accessed the variable in the translators for event and enroll.

```
message.getProperty("CDAVAR", PropertyScope.INVOCATION);
```

Listing 6.3: Using the Mule Flow Variable.

The first custom Java translator, Cda to DHIS2 Person, received a POCDMT000040-ClinicalDocument Java object and mapped the relevant information to a JSON format that can be used to create a person in DHIS2.

Different encoding of organization units in SIVIGILA and DHIS2 presented a challenge. Its solution is described in section 6.7 on page 92. DHIS2 organization units UIDs were required to insert persons in DHIS2. Thus a data dictionary was created. This and other data dictionaries were hard coded into the translator.

I used a JSON template extracted from the DHIS2 REST API to add the values to the correct attributes. The software package json-simple was used for the mapping and creation of the JSON message. Json-simple is a Java toolkit for JSON (*json-simple* n.d.). Maven support in Mule Studio made it easy to setup json-simple for the Mule project. The DHIS2 REST API provided the named template and important information such as organizational units, identifiers and relevant attributes.

I used a Hashmap to transfer the information from the structured body section in the POCDMT000040-ClinicalDocument object into the values and attributes of the JSON message.

The development work used a DHIS2 database already containing the information in the SIVIGILA input file. So I was forced to add logic to create new ID numbers. This logic must be removed when using this solution.

The JSON object with all information was then sent to the Person to JSON translator element. It serialized the JSON object to a JSON message.

The HTTP endpoint Post Person to DHIS2 element posted the JSON message to the REST DHIS2 Person service, dhis/api/persons, using a request-response pattern and basic authentication. The response was then processed as decribed in Part Two.

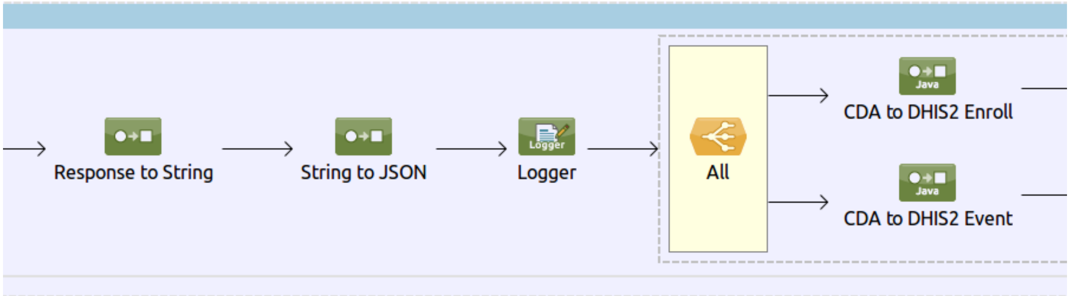


FIGURE 6.9: DHIS Mule Adapter Part Two.

6.5.4 Part two

The response from the Person REST API was translated into a String and logged as shown in figure 6.9. The Logger element forwarded the response to a flow control element that routed the message into two routes: one for enrolling a person and the other to create a program event.

The first route started a custom Java translator named CDA to DHIS2 Enroll. It merged the response from the Person REST API with the CDA flow variable creating a new enrollment JSON object to be used in Part Three.

The second route started with a custom Java translator named CDA to DHIS2 Event. It merged the same information as the first route to create an event JSON object to be used in Part Three.

6.5.5 Part Three

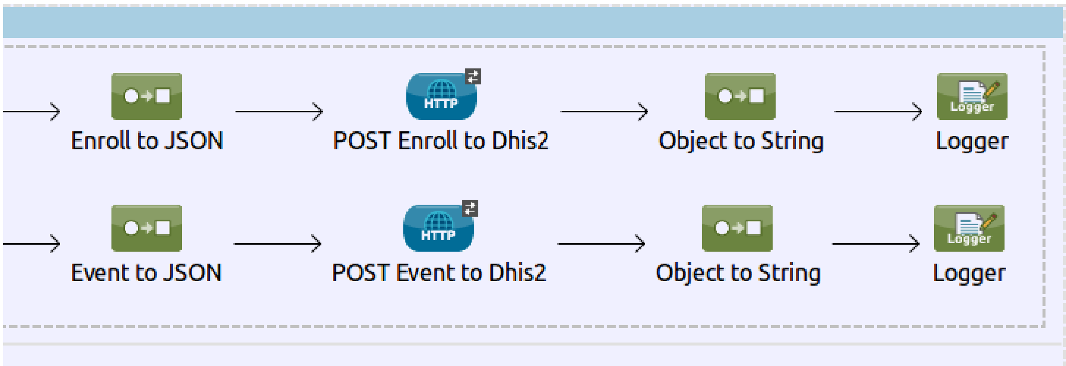


FIGURE 6.10: DHIS Mule Adapter Part Three.

Route one in part three (the upper route in figure 6.10) continued the first route by serializing the JSON Enroll object to a JSON message in the Enroll to JSON element.

The Post Enroll to DHIS2 HTTP endpoint posted the JSON message to the DHIS2 REST service dhis/api/enrollments. The response was sent to an 'Object to String' translator and then logged.

Route two continued by serializing the JSON Event object to a JSON message in the Event to JSON element. The POST Event to DHIS2 HTTP endpoint posted the JSON message to the REST DHIS2 service dhis/api/event. The response was then translated to String and logged.

6.6 Using the CDA as a CDM

There were several challenges that were encountered when trying to use the CDA schema (XSD) as a CDM. The 3 main challenges were binding the schema, making it serializable and using mixed content elements. These issues arose due to the way the CDA XSD is written.

The task of binding an XSD involves the automatic generation of Java classes where each Java class corresponds to a XML element and each Java field corresponds to an XML attribute. The binding process also generates factory classes to facilitate object creation. The generated Java classes are often referred to as POJOs or Plain Old Java Objects.

To bind the CDA XSD and generate POJOs I created a maven project. It generated Java POJOs by executing XJC and used JAXB bindings. The result was a set of POJOs that I could compile and export as a Java library. It was included as a dependency in the Mule adapter projects. At first using JAXB with XJC was not successful. This is due to the fact that the XSD has circular dependencies. To fix this issue I used inline JAXB bindings. One example of an inline binding for the solution is shown in listing **??** on page ??. XJC was executed successfully with the bindings.

```
<xs:attribute name="ID" type="xs:ID"> <xs:annotation>
        <xs:appinfo>
          <jaxb:property name="ID1"/>
        </xs:appinfo>
    </xs:annotation>
</xs:attribute>
```

Listing 6.4: JAXB Binding One

However a second issue arose because the ActiveMQ (MOM) required the CDA object to be serializable and gave an error when I used the unserialized objects. Thus I had add another binding to solve this issue. Listing 6.6 on the next page shows how the problem was solved.

The third problem arose with the use of mixed content for HL7 data types such as ED and ST. It was not possible to use Title (ST) or text (ED) elements due how XJC dealt with mixed content. It lead to some confusion on how I could set ST and ED according to CDA. I later discovered that it was because of the use of mixed content, an example of this is shown in listing 6.5. To overcome this problem I used a JAXB binding as shown in listing 6.6. This meant that the CDA could now be used as level 1 and could use the mentioned data types.

```
<xsd:complexType name="ED" mixed="true">
```

LISTING 6.5: Mixed Content Problems.

```
<xs:annotation>
    <xs:appinfo>
      <jaxb:globalBindings generateMixedExtensions="true">
        <xjc:serializable uid="12343" />
      </jaxb:globalBindings>
    </xs:appinfo>
  </xs:annotation>
```

LISTING 6.6: JAXB binding two and three.

The Second Implementation used inline bindings, but one could use a separate binding file. It would make it easier to maintain them.

6.7 Encountered Problems

6.7.1 More Semantics More Problems

There was no authority that defined shared codes for general interoperability of Health Information Systems in Colombia. This resulted in SIVIGLA and DHIS2 mostly defining their own codes independently of each other. The codes in SIVIGILA were mainly numbers and abbreviations. The codes in DHIS2 were mainly meaningful words in Spanish meant to be presented to the end user. Under these circumstances I decided to use the available SIVIGILA codes in the CDM.

The Second Implementation included some hard coded dictionaries in the translators. Such dictionaries covered ethnicity, type of ID document, health regimen, type of case, area and others. Listing 6.7 on the following page shows how I solved one of the issues by hard coding a data dictionary in a translator. The values in the CDA for ethnicity are number one to four. The method injectGropuoEntnicoFamiliar() writes the meaning of the codes for ethnicity in a JSON format that would later be transferred to DHIS2.

Other dictionaries used external mapping files due to their size. They were dictionaries for organization units, SIVIGILA events, ICD-10, countries, municipalities, villages and others.

```
private void injectGrupoEtnicoFamiliar(JSONObject jsonTemp, HashMap<String,String> siviMap) {
    if(siviMap.get("Etnia") != null){
      switch (siviMap.get("Etnia")) {
        case "1":
          jsonTemp.put("value", "1 -Indigena");
          break;
        case "2":
          jsonTemp.put("value", "2 -Rom(gitano)");
          break;
        case "3":
          jsonTemp.put("value", "3- Raizal del archipielago de San Andrús y
          Providencia");
           break;
        case "4":
          jsonTemp.put("value", "4- Palenquero de San Basilio");
          break;
        case "5":
          jsonTemp.put("value", "5- Afrocolombiano,Mulato,Afrodescendiente");
          break;
        case "6":
          jsonTemp.put("value", "6 - Otros");
          break;
        default:
          break;
      }
    }
    else{
      jsonTemp.put("value", " ");
    }
  }
```

Listing 6.7: Semantical Solution. Hard coded Data Dictionary.

Due to time constrains I did not create a data dictionary for occupation that contains the SIVIGILA code as key and name as value. This information is available in DHIS2 Cauca database.

It should be noted that some codes used in the data dictionaries, were found in the DHIS2 database because I did not have access to official SIVIGILA code lists.

A related problem was identifying the SIVIGILA column names. SIVIGILA used an abbreviation for its column names. I converted the SIVIGILA column name abbreviations to a meaningful Spanish name in the CDM. e.g a SIVIGILA column name *per_ etn* was converted to *Etnia* in the SIVIGILA Mule adapter.

The next problem I faced was the CDA itself.

6.7.2 CDA problems

As mentioned in section 6.6 on page 91, there were issues using the CDA as a CDM due to how it is written. In addition the used documentation was spread out and difficult to understand. The RMIM was used for getting an overview. The HMD provided more details but lacked some elements. Browsing the XSD schema in eclipse gave the required detail but was time consuming due the CDA complexity.

I also had to use Benson (2012), Boone (2011) and the official CDA documentation to understand and select CDA elements. Benson (2012) provided a broad overview while Boone (2011) provided more detail and examples. However, the latter contained some errors and did not cover all the details. The official documentation was detailed but written in a hard to understand language and it was easy to 'get lost' reading it.

Both getting an overview and understanding CDA proved challenging and time consuming due to CDAs complexity. I communicated via email with Grahame Grieve a HL7 expert and the leading architect of the emerging HL7 standard FHIR to solve some doubts.

6.7.3 DHIS2 API Documentation

This interoperability solution was developed for version DHIS 2.14. Version 2.15 was released in Spring 2014 and the documentation for version 2.14 was changed to reflect the new version. This resulted in documentation labeled as 2.14 with content describing version 2.15. The API for creating Person was thus wrongly documented as replaced by Tracked Entity. I had to use unnecessary time sorting out the differences between the documentation and the version I was using. The developer community answered that tracked entity was not available in version 2.14. I had to resort to google to find cached documentation.

6.8 CDM Conversion Details

This section presents the conversion details of the CDA. The chosen parts are shown in XML code that was marshalled from the Java CDA objects during testing. The CDA objects are sent to the message bus and later translated to a DHIS2 format as shown in section 6.4 on page 85. The important CDA parts are clinical document, participants and body.

An overview of the CDA R-MIM can be seen in appendix F on page 149. The CDA figures in this section are taken from the R-MIM. I used the CDA HMD (in the same appendix) to select the relevant sections.

6.8.1 Clinical Document

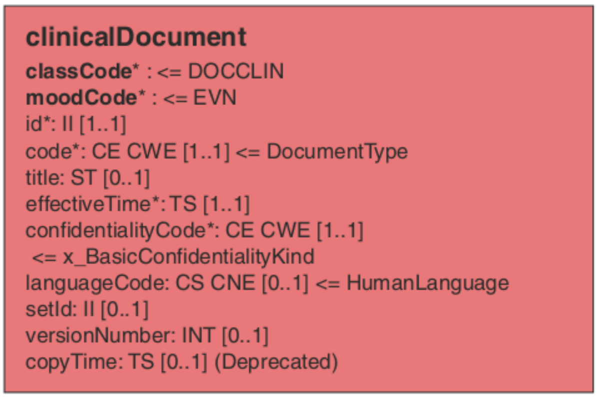


FIGURE 6.11: Clinical Document (Boone 2011).

The clinical document header in figure 6.11 shows different elements and attributes, their data type and cardinality. Listing 6.8 shows how I injected title of the clinical document in the SIVIGILA adapter. Listing 6.9 on the following page show a marshalled XML snippet for clinical document. I did not focus on converting the time format that SIVIGILA uses to a CDA ISO format because of time constrains.

```java
private void injectTitle(POCDMT000040ClinicalDocument clindoc){
    ST st = new ST();
    List<Serializable> stList = new LinkedList<Serializable>();
    stList.add("Datos Basicos");
    st.setContent(stList);
    clindoc.setTitle(st);
  }
```

LISTING 6.8: Inject Title to CDA

The code element specifies the particular kind of document with content from LOINC [§4.2.1.2]. The LOINC code for Public health case report was used, and the language code was set to Spanish (Colombian). The mood code is set to the fixed value EVN. EVN is defined as: "An actual occurrence of an event (i.e., the documentation act already happened and is not just a request, intent, plan or promise to document). " [§4.2.1]

The id represents the unique instance identifier of a clinical document. How this should be defined is not up to me to decide, so for the time being it is a fixed value.

The confidentiality code is set to N, that is Normal. There are other options such as: R (restricted) and V (very restricted). Normal is defined as “Normal confidentiality rules (according to good health care practice) apply. That is, only authorized individuals with a legitimate medical or business need may access this item” [§4.2.1.5]. It should be noted that I have no authority to define the access level. Guidelines for document access should be defined by a relevant national or departmental authority.

The effective time is used as document creation time [§4.2.1.4]. I could use a java function that was used in another translator to set the system time. However as the document was supposed to represent the SIVIGLA data form, I used the time created from the CSV file.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ClinicalDocument xmlns="urn:hl7-org:v3">
    <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
    <id extension="c266" root="2.16.840.1.113883.1.3"/>
    <code displayName="Public health case report" codeSystemName="LOINC↩
    " codeSystem="2.16.840.1.113883.6.1" code="54217-5"/>
    <title>Datos Basicos</title>
    <effectiveTime value="2/24/2013"/>
    <confidentialityCode codeSystem="2.16.840.1.113883.5.25" code="N"/>
    <languageCode code="ES-CO"/>
```

Listing 6.9: Clinical Document part of a Marshalled CDA During Testing.

6.8.2 Participants

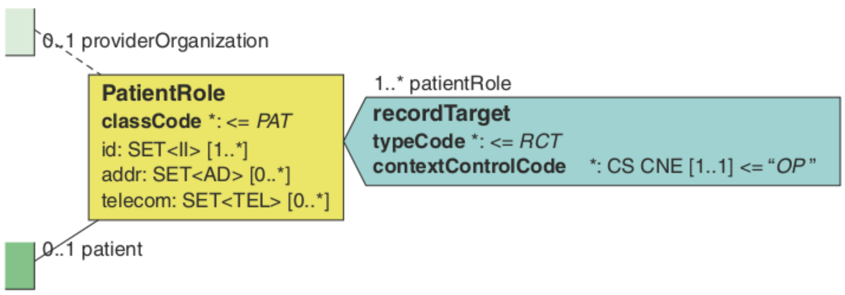


Figure 6.12: Record Target and Patient Role (Boone 2011).

There are several participants in a CDA. This section will show the participants that were created. Participants have a blue colour in the R-MIM. Figure 6.12 shows the **record target** participant. The record target “represents the medical record that this document belongs to” [§4.2.2.11]. Thus it is represented as a relationship between a person and an organization [§4.2.2.11]. Listing 6.10 on the following page shows the marshalling of a translated CDA object during testing. The presented information is anonymized. The context control code is set to OP, and the type code to RCT [§4.2.2.11].

The patient role is also shown in figure 6.12 on the previous page and its marshalling in listing 6.10. The patient role class "represents the person playing the role of patient" the address and telcom elements represent the address and telephone numbers by the person in a given patient role (Boone 2011, p.150). The telephone number was injected from the SIVIGILA CSV; the address could have been injected but due to lack of time this was not done.

```
<recordTarget contextControlCode="OP" typeCode="RCT">
        <patientRole classCode="PAT">
            <id extension="55140250" root="2.16.840.1.113883.19.5"/>
            <telecom value="tel:8284627"/>
            <patient determinerCode="INSTANCE" classCode="PSN">
                  <name>
                    <given>name</given>
                    <given>name</given>
                    <family>name</family>
                    <family>name</family>
                </name>
                <administrativeGenderCode codeSystem="↩
    2.16.840.1.113883.5.1" code="F"/>
                <birthTime value="7/11/2007"/>
            </patient>
        </patientRole>
    </recordTarget>
```

Listing 6.10: Record Target part of a Marshalled CDA During Testing. Values are anonymized.

The patient class is shown in figure 6.13 on the next page. It represents an entity (noun) used in clinical statements. This class describes "person who is the patient in the context of the clinical document" (Boone 2011, p. 161). To get the name and given tags right I faced an issue due to the fact that name had a Serializable list. Thus I had to create a method that could add the correct format to the list. This was done with creating a JAXB element because the object factory was lacking this feature. I used the method shown in listing 6.11 with "given" and the corresponding name from the SIVIGILA CSV. For the last names I used the prefix "family"

```
private JAXBElement<String> createJaxbElement(String prefix, String ↩
    name){
    QName q = new QName("urn:hl7-org:v3",prefix, name);
    return new JAXBElement<String>(q, String.class, name);
  }
```

Listing 6.11: Creating a JAXB Element.

The id is deprecated in CDA R2 due to the fact that a patient does not have a single id (Boone 2011). The classcode is set to PSN. PSN is defined as "A living subject of the species homo sapiens." [§4.2.2.11]. The determiner code chosen is the default INSTANCE

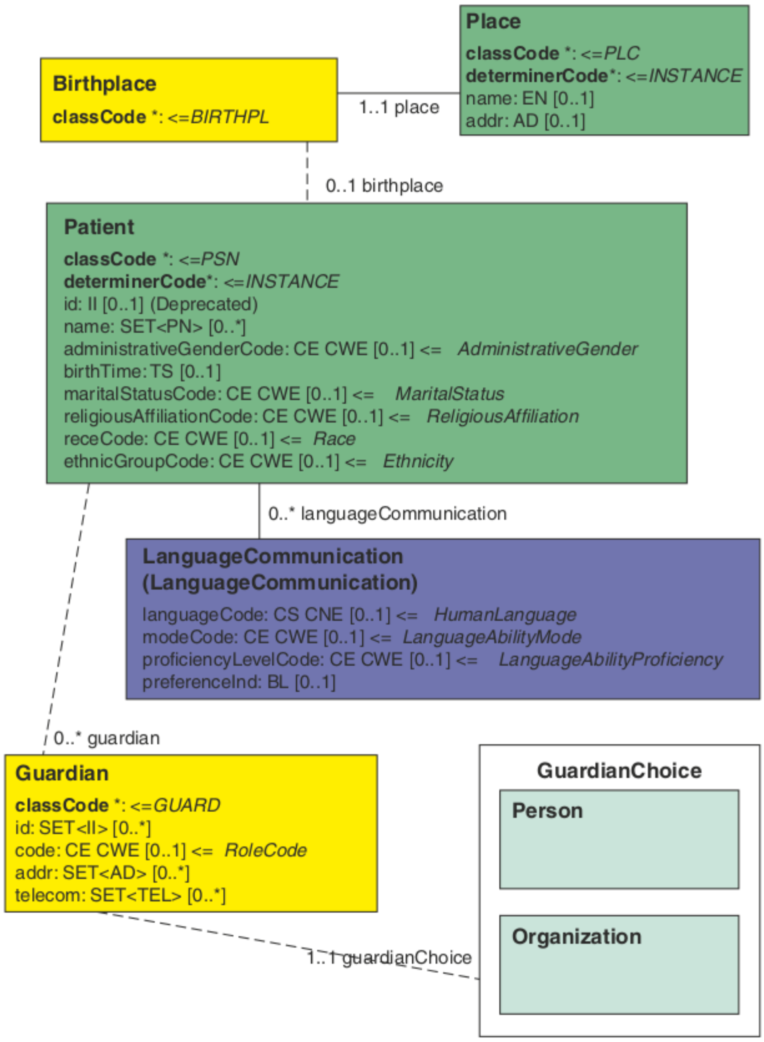


Figure 6.13: Patient (Boone 2011).

that can be defined as "The INSTANCE determiner indicates an actual occurrence of an entity" [§4.2.2.11].

The elements that are injected to the patient are name, administrative gender code and birth time. I could map more such as race code but due to time it was not done.

The gender code represents the gender of a given patient for administrative purposes. Because using a clinical description would include "anatomical and hormonal gender" (Boone 2011, p. 162)

The name element describes names for the patient. There are many types of possible names that could be used but I am mapping the legal names. The birth time describes the time of birth for the patient. It was also mapped from the SVIGILA CSV file.

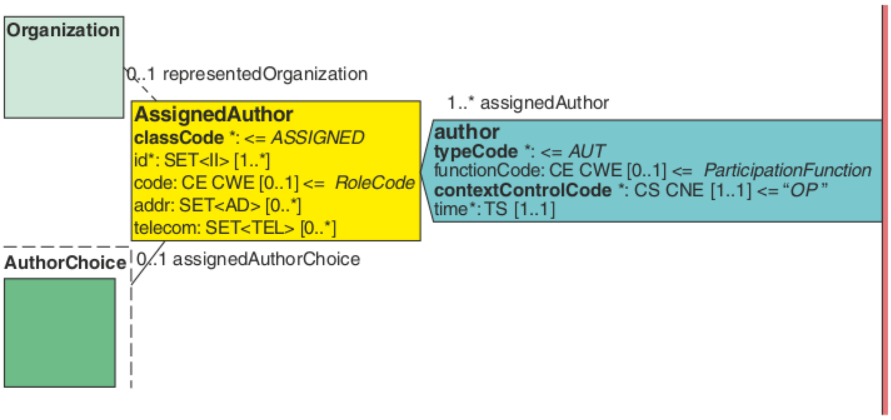


FIGURE 6.14: CDA Author (Boone 2011).

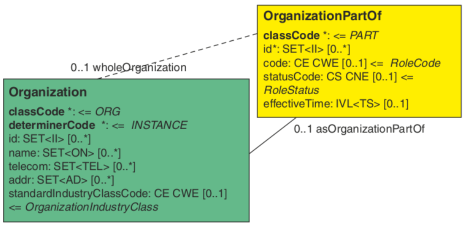
The **author** class is shown in figure 6.14 and it "represents the humans and/or machines that authored the document" [§4.2.2.2]. It can be a person or a device and is a required element in the CDA. The authors "create information in the clinical document based on their knowledge or application of skills" (Boone 2011, p. 150).

Listing 6.12 shows the marshalled code of the author Java object. The author context control code is set to OP (overriding propagating), that is "The participant overrides associations with the same typeCode. This overriding association will propagate to any descendant Acts reached by conducting ActRelationships"[§4.2.2.2]. The type code AUT stands for "A party that originates the Act and therefore has responsibility for the information given in the Act." [§4.2.2.2]. Time value is an example value as I had no information on the value from the SIVIGILA input file.

```
<author contextControlCode="OP" typeCode="AUT">
    <time value="20130203001746"/>
    <assignedAuthor classCode="ASSIGNED">
        <id extension="KP00017" root="2.16.840.1.113883.19.5"/>
        <assignedPerson determinerCode="INSTANCE" classCode="PSN">
            <name>
                <given>Jaime</given>
                <family>Loza</family>
                <suffix>MD</suffix>
            </name>
        </assignedPerson>
        <representedOrganization determinerCode="INSTANCE" classCode="ORG"/>
    </assignedAuthor>
</author>
```

LISTING 6.12: Author Part of a Marshalled CDA During Testing.

The assigned author is required as shown in figure 6.14. I used the class code ASSIGNED for the assigned author. ASSIGNED is defined as "A role in which the playing entity is



(a) CDA Organization (Boone 2011)

(b) CDA Person (Boone 2011)

FIGURE 6.15: Organization and Person.

acting in the employ of or on behalf of a scoping organization" [§4.2.2.2]. The id element identifies the author "but not the same value as the identifier used by authoring persons to access the system creating the document." (Boone 2011, p. 152). I used an example id because there were no guidelines defining which id should I should chose.

The assigned author can have an assigned person. Figure 6.15(a) shows the person class in the R-MIM. The determiner code is set to INSTANCE and class code to PSN, as in the record target patient. The Person can have 0 or more names. In the case of SIVIGILA it would be the name of the professional that filled the form. This is not mapped but it could be mapped from the SIVIGILA CSV. The code was not used but it could be done if information is available it would specify the level of education of the health care provider. (Boone 2011, p.152). The telephone number could also be set by using the CSV.

There is an option to choose a represented organization for an assigned author. Figure 6.15 shows the organization class. It has a class code of type CNE that is ORG. ORG is defined as "A social or legal structure formed by human beings." [§4.2.2.2.3].

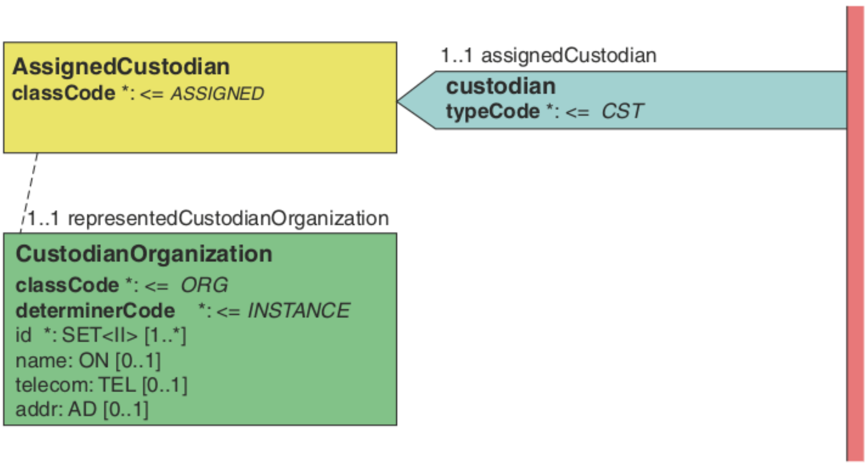


FIGURE 6.16: Custodian (Boone 2011).

The clinical document must have one **custodian**. Figure 6.16 on the preceding page shows the the custodian in the R-MIM. The custodian “represents the organization that is in charge of maintaining the document” [§4.2.2.3]. The type code I used for custodian is CST: “An organization that is in charge of maintaining this document” [§4.2.2.3]. It is up to the organization to decide how long the document would be maintained (Boone 2011). It must have one assigned custodian, the class code I used for the assigned custodian is ASSIGNED. Listing 6.13 shows the marshalled code for custodian. The custodian should be SIVIGILA but due to time constrains it was left out.

```
<custodian typeCode="CST">
      <assignedCustodian classCode="ASSIGNED">
          <representedCustodianOrganization determinerCode="INSTANCE"
   classCode="ORG"/>
      </assignedCustodian>
</custodian>
```

Listing 6.13: Custodian Part of a Marshalled CDA During Testing.

6.8.3 Body

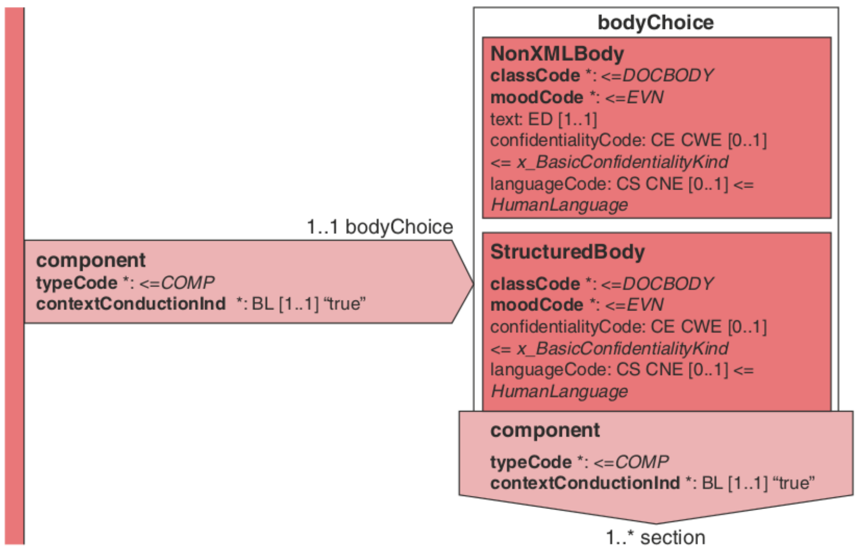


Figure 6.17: Body and Components (Boone 2011).

The **body** of a CDA can be either a non XML body or a structured body as shown in figure 6.17. During my initial implementation I tried to use the nonXML body. However I encountered the issue of not finding where to inject the content in the object as stated in section 6.6 on page 91 due to the mixed content issue. Using the nonXML body would make the CDA a level one as shown in chapter2. As shown below it was only too late that I found out a solution for accessing the mixed content in this element.

Rather than wasting time on trying to find out why I could not set the content I started using a StructuredBody. This would allow me to set content from the CSV file. However, this approach also have difficulties.

The StructuredBody must have a class code and a mood code. Listing 6.14 shows the marshalled code for the StructuredBody I used. Some items are replaced by [...etc] to make it shorter. The moodCode EVN and classCode is set to DOCBODY that is defined as “A context that distinguishes the body of a document from the document header.” [§4.3.1.1] The confidentiality code was not set as it is optional and I do not have the authority to do so.

```
<component contextConductionInd="true" typeCode="COMP">
       <structuredBody moodCode="EVN" classCode="DOCBODY">
           <component contextConductionInd="true" typeCode="COMP">
               <section moodCode="EVN" classCode="DOCSECT">
                   <title>Ficha De Notificafion - Cara A</title>
                   <text>
                       <list>
                           <item>Codigo Departamento Occurencia:19</↩
  item>
                           <item>Sexo:F</item>
                           <item>Codigo Evento:300</item>
                           <item>Certificado de defuncion:</item>
                           [....etc]
                       </list>
                   </text>
               </section>
           </component>
       </structuredBody>
 </component>
```

Listing 6.14: Body Part of a Marshalled CDA During Testing.

To use the StructuredBody I had to create a component2 that is required by the clinical document. Component2 has a type code COMP and the contextconductionInd element is set to true. COMP is defined as “The associated document body is a component of the document ”[§4.3.1]. To use a section I had to create a component3 that can have one or more sections.

Note that the figure taken from Boone (2011) contains an error as there is no type code in the section class but rather a class code that is equal to DOCSECT.

Figure 6.18 on the following page show the section that is defined using the id element. The mood code was set to EVN and class code to DOCSECT. DOCSECT is defined as “A context that subdivides the body of a document” [§4.3.2]. A LOINC code could be used as a code element but I was not sure which code was the correct one to use. One

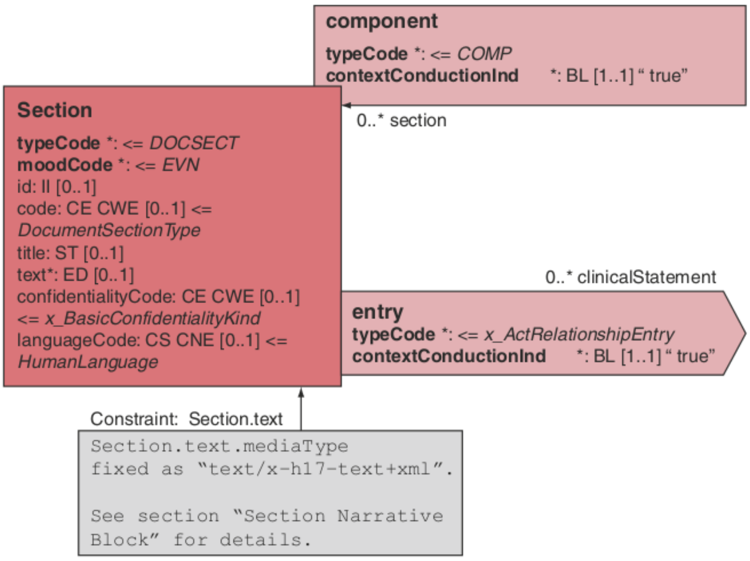


FIGURE 6.18: Section and Component (Boone 2011).

problem encountered when using the section was injecting the SIVIGILA values to the sections text element (strucDocText).

I met a compound problem of nested lists with two circular references in different parts of the nesting and elements with mixed content. Figure 6.19 on the next page[1] shows strucDocText, List and Item. Figure 6.20 on page 105 shows the problem with strucDocContent due to recursive references. The problem first arose because I could not set StrucDocText elements such as StrucDocList. When the problem with content was fixed I used StrucDocList in strucDocItem and its content to set data from SVIGILAs CSV. Listing 6.16 on the next page show how I solved it in the SIVIGILA adapter.

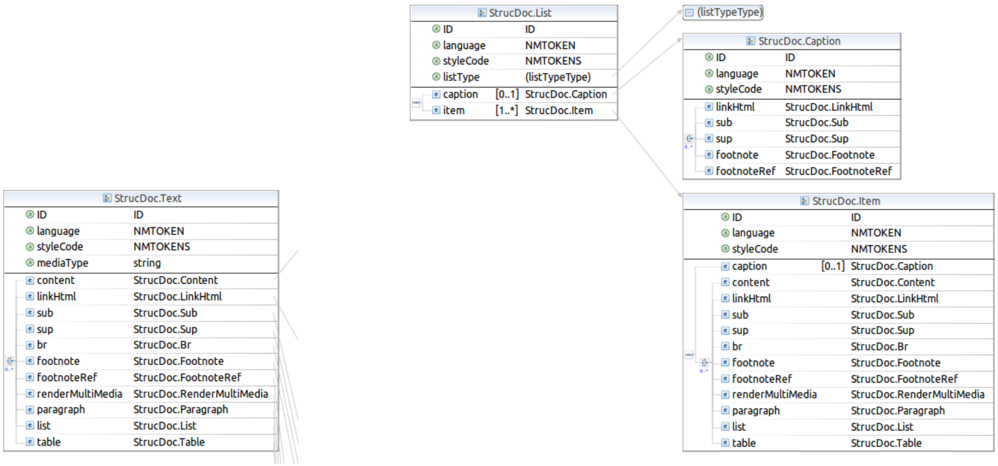
```java
private JAXBElement<StrucDocList> createJaxbElementList(StrucDocList strcdcList, String prefix){
  QName q = new QName("urn:hl7-org:v3", prefix);
  return new JAXBElement<StrucDocList>(q, StrucDocList.class, strcdcList);
}
```

LISTING 6.15: Creating a JAXB Element for StrucDocList.

To solve this compound problem I also used an instance of JAXBElement for StrucDocList. Listing 6.15 show the method that was used for creating the instance.

The section title was first not set due to the issue with mixed content as shown in section 6.6 on page 91. This issue is not due to data type ED but rather ST (standard

[1]The figures are split in two sub figures because eclipse cannot show them in one figure.



(a) Struc Doc Text. (b) Struc Doc List and Item.

FIGURE 6.19: StrucDocText and StrucdocList, StrucdocItem.

text). When I fixed the issue I injected the correct data to a content list in the BIN class. (ST and ED extends BIN)

```
private void injectSectionText(StrucDocText strucDocText, POCDMT000040Section sectionInfGeneral){
  StrucDocList strlist = new StrucDocList(); LinkedList<StrucDocItem> items = new LinkedList<StrucDocItem>();
  LinkedList<Serializable> txtContent = new LinkedList<Serializable>();
  for (String name : sivigilaDataMap.keySet()) {
    String key = name.toString();
    String value = sivigilaDataMap.get(name).toString();
    StrucDocItem strItemX = new StrucDocItem();
    LinkedList<Serializable> strItemListX = new LinkedList<Serializable>();
    strItemListX.add(key + ":" + value);
    strItemX.setContent(strItemListX);
    items.add(strItemX);
  }
  strlist.setItem(items);
  JAXBElement<StrucDocList> jaxListStr = createJaxbElementList(strlist, "list");
  txtContent.add(jaxListStr);
  strucDocText.setContent(txtContent);
  sectionInfGeneral.setText(strucDocText);
}
```

LISTING 6.16: Setting StrucDocItemContent.

6.9 Scope

The Second Implementation is a prof of concept solution. It leaves aside the response track in the message exchange, transactional support and the conversion of some elements: It is not fit for deployment. Testing was limited to manual control that records

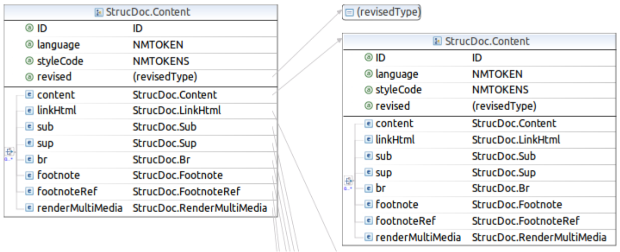


FIGURE 6.20: Recursive StrucDoc.Content.

in the input file resulted in correct changes in the DHIS2 instance.

6.10 Summary

This chapter described the second interoperability implementation and the tools used in its development. It differentiates itself from the first implementation by using an ESB (Mule and ActiveMQ). Details on two Mule adapters are shown and challenges related to using CDA are described[2]. Challenges with organizational deficiencies establishing a shared interoperability platform with code sets, organizational structures and implementation guidelines were also described. Implementation challenges with spread and partially inaccurate documentation were also presented. The CDA complexity lead to the discovery of the mentioned organizational deficiencies pointing to a mismatch between the project ambitions and the means to fullfill them. The next chapter uses the presented information for discussion purposes.

[2] There was no time left to clean up the custom Java translators partially due to time limitations and also due to personal health concerns.

Chapter 7

Discussion

This chapter presents in three sections the discussion of the empirical work from chapter 4, 5 and 6. It answers the research questions as described in chapter 1. Section 7.1 analyses the three relevant architectural approaches and the technical work for both implementations. Section 7.2 shows an analysis based on the three levels of interoperability and how they influence each other. I finally present reflections on my research approach in section 7.4.

7.1 Reflections on the Technical work

The technical work faced the type of challenges described by (Erl 2004) in terms of having distributed heterogeneous systems that require interoperability. Other challenges as shown by Benson (2012) and Hohpe & Woolf (2003) were also encountered. The challenges presented by the latter were the use of integration patterns for creation of loosely coupled systems and the fact that systems are always changing. While challenges described by the former were encountered du to issues regarding development in terms of knowledge and communication.

The three main architectural approaches: Point to point, Hub and Spoke and Enterprise Service Bus (ESB), as shown in chapter 2, were encountered in the field work.

Finally a Canonical data model (CDM) was used to create loose coupling and could in theory reduce the number of required interfaces.

7.1.1 Architectural Approaches

7.1.1.1 Point to Point

The already developed interoperability solution for SIVIGILA and DHIS2 at the time the project began was a point to point database integration. Hohpe & Woolf (2003) argues that this kind of integration can be problematic as systems can be tightly coupled and can create 'spaghetti integration' as shown in chapter 2.

After the failure of the First Implementation the SDSC IT team used Talend, an ETL tool with a point to point solution between SIVIGILA and DHIS2. Further work with this type of interoperability to cover all systems in Cauca would require $(30^2 - 30)/2 = 435$ interfaces if all the systems would need to be interoperable with each other. This scenario would lead to what Manouvrier & Ménard (2008) calls *fossilisation* and tight coupling. This interoperability solution would however support this scenario of only two intervening systems and another one directional scenario.

The other scenario is for transferring data from 30 systems just to DHIS2. It would require 30 different interfaces translating from the given system format to DHIS2s format. This scenario is best attained if the DHIS format remains static over time. Whenever DHIS2s would update its format, all 30 interfaces would need to be updated in order to maintain interoperability.

30 systems would in total require 30 translators. If DHIS2 would change its API, like was the case from 2.14 to 2.15 where Tracked Entity was introduced, then all 30 translators would require an update. Thus having a point to point interoperability solution with DHIS2 would require a high amount of maintenance work.

If the point to point solution would to be developed with a database integration it would require even more work than using the API. Each translator would then depend on DHIS2 internal logic. Changes to DHIS2 internal data model could impact all 30 translators and a higher degree of adaptation would be required to update the translators.

Figure 7.1 on the following page shows a point to point interoperability scenario in a component diagram with four systems: SIVIGILA, RIPS, RUAF and DHIS2. It covers the scenario of having many systems transferring data to DHIS2 without being interoperable with each other. A translator would be needed for each system that would be interoperable with DHIS2. Making a new system interoperable with DHIS2 would require a new translator for the given system. They would require a translation from the system format to a DHIS2 format.

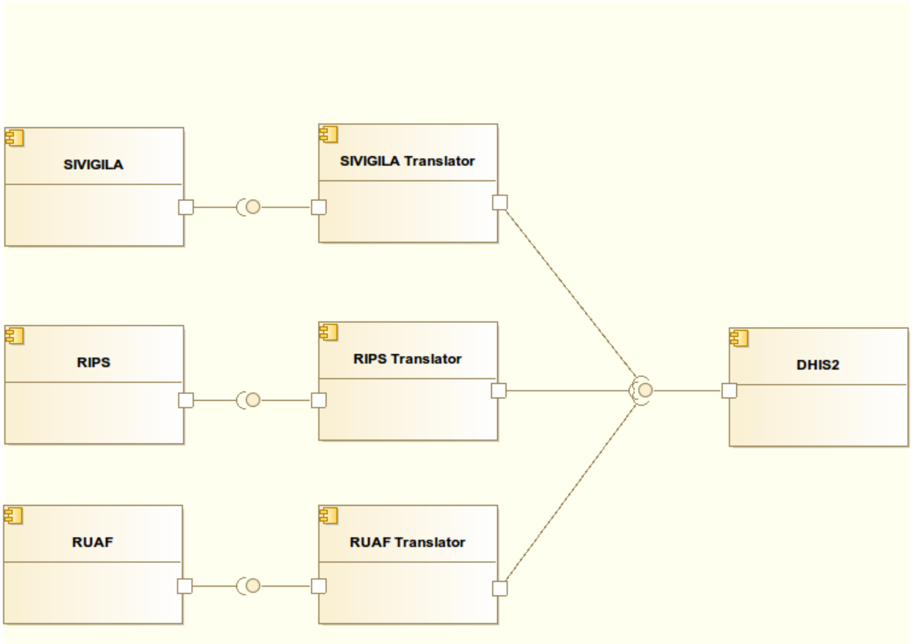


FIGURE 7.1: Point to Point integration with three Systems.

In conclusion there were two different point to point integration solutions that were used during my work in Cauca without my participation. Both would produce tight coupling and a degree of *fossilization* and would reduce flexibility, agility for all three levels of interoperability.

Furthermore, systems could be be added as the Cauca Information Infrastructure[1] evolves. Fossilisation and tight coupling would make the systems change resistant within the II. Adding one more system to the interoperability solution would require 30 interfaces to connect it to the other systems making in total 465 interfaces. Adding 10 more systems would require in total 780 interfaces. The tight coupling means that change in one system could affect all other systems. In sum it would lead to *inert* evolution: the II cannot add or change systems without introducing a high degree of complexity.

7.1.1.2 Hub and Spoke

The Hub and Spoke envisioned solution with MIRTH Connect in the first implementation (chapter 5), would reduce the amount of translators compared to the point to point solution because it has a CDM. The maximum number of translators would only be 30 also to cover general interoperability (first scenario). Thus adding new systems to the solution would always require x translators for x systems.

[1]The Cauca II is based on heterogeneous elements of technical, institutional, social, practices. This constitutes the install base that enable and can restrict evolution.

However, the main problem with the Hub and Spoke architecture would not arise with few systems but rather when the solution has to be scaled (Erl 2004). The scaling could be a problem as all transformations are performed in a central hub that could quickly become a bottleneck.

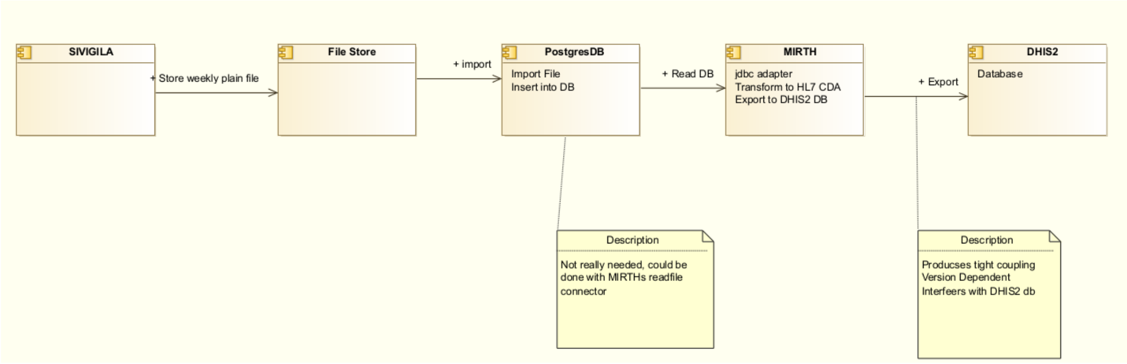


Figure 7.2: Interoperability weaknesses for a Hub and Spoke Approach.

According to Erl (2004, p.372) the bottleneck situation is a danger "unless scalability is carefully planned and designed into the hub infrastructure". To face the challenge of having a central hub being a single point of failure one could take measures such as replication and clustering. However "the cost of putting together a fully scalable hub and spoke environment with full fail-over support may be prohibitive." (Erl 2004, p.372)

Thus a hub and spoke solution would be costly in the long term to the point of being not only a technical bottleneck but also be restricted by lack of available funds.

While a Hub and spoke is less complex than an ESB, it would not be the case when one uses MIRTH Connect (with limited funds) because of the lack of access to CDA API and documentation. The fact that one wanted to use such a tool without the funds for support or access to the CDA shows organizational weaknesses and inexperience. It reflects the challenges mentioned by Benson (2012): Users don't understand the development process and developers don't know the domain. Consequently, the solution contained several weak points.

Figure 7.2 shows a Postgres database used to merge two files. It duplicates the effort without giving any advantages. The direct insert to the DHIS2 database creates close coupling and makes the interoperability solution version dependent. If a new version of DHIS2 changes its database structure it forces adapting interoperability logic to change.

Additionally, the chosen canonical data model was a CDA message template rather than the whole CDA schema. This requires less effort and complexity in the beginning but as the integration solution grows, different systems can have different template versions for sending data. It defeats the purpose of the CDM. These problems are addressed with an ESB solution.

This approach only needs x interfaces for x systems when systems are added given the evolution of Cauca II. Thus adding 10 systems only means having in total 40 interfaces compared to the point to point approach that entails 465. Loose coupling would be achieved by this approach making the evolution more *active* and adaptive.

7.1.1.3 ESB

The ESB solution (chapter 6 on page 79) is shown in figure 7.3 with SIVIGILA, RIPS, RUAF and DHIS2. It also shows the different adapters that are required for the different systems in the interoperability solution. The figure shows how the three systems would transfer data to DHIS2 and between them. The message bus could send CDA messages between any adapters.

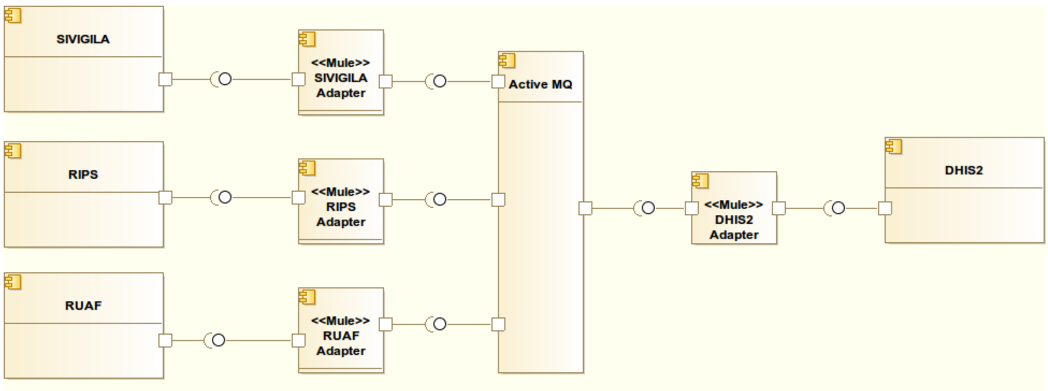


FIGURE 7.3: ESB Interoperability with three systems.

The fact that the solution would be distributed rather than having a central Hub would make it more scalable and reduce cost compared to the Hub and Spoke architecture with Mirth Connect. At the same time the benefits would mainly be shown when more systems used the ESB by adding adapters or when one or more systems updates to newer versions.

One weakness that could develop is the fact that the dependencies that would be clear in a point to point solution would be 'forgotten' and thus the complexity would lead to hidden dependencies. One would have to know what one adapters modification would entail for the dependent systems. Thus "High interoperability must be accompanied by a well-defined architecture, structures, and processes" (Josuttis 2007, p.7). They were lacking for the two interoperability solutions.

The solution lacks some important parts. Firstly, it is based on a 'happy day' scenario where there are no errors. Time constrains shaped the scope of this result. Some errors are logged but no reaction is provided. This is because the scope is only for fire and forget and there is no way to return an error message.

The transactional rollback is also missing, if a person is created but there occurs an error in the event or enroll phase the person is not deleted. A rollback should also be done if the person and enroll phase is ok but the event fails.

This solution uses Mule. One of Mule's strengths is that it uses the integration patterns Hohpe & Woolf (2003) and has a graphical flow. Other aspects of an ESB implementation were not taken into account such as: security and deployment.

As the Cauca II evolves the ESB solution would provide a loosely coupled solution. X added systems requires x interfaces. Thus adding 10 systems to the solution requires in total 40 interfaces, similar to the Hub and Spoke approach. The difference being that the ESB is a distributed solution rather than a centralized one. This removes the potential bottleneck problems faced by a Hub and Spoke approach. A similarity to the latter is that the ESB approach also leads to *active* evolution. However, the lack of well defined architecture, structure and process can lead to hidden dependencies. Thus, while the ESB solution can enable *active* evolution, it can also lead to implosion of the interoperability solution by using a 'revolutionary' like approach, resulting in *slow* evolution.

7.1.2 Discussion in terms Architectural Approaches

Table 7.1 presents the different strengths and weaknesses for the implementations[2]. P2P relates to the solution that the SDSC IT team developed for the DHIS2 pilot. The Hub and Spoke (H&S) approach to the first implementation with Mirth Connect and ESB with the second Approach with Mule.

The P2P approach, compared to the others, was simpler and easier to use. It did not use a CDM. Thus the knowledge required to develop it was much smaller. It worked for the pilot project of DHIS2. Its weaknesses can emerge with many systems because it creates tight coupling, meaning that each system needs an interface for every connection. Its lack of complexity meant that it was quick and simple to develop for the pilot. The P2P approach could be appropriate for pilots similar to Cauca's pilot and interoperability scenarios with few systems.

[2] It should be noted that the discussion is limited because of the project scope. It does not take into account fully working and deployed solutions. In addition, the chosen product for each approach strongly influenced its strengths and weaknesses. This makes the discussion highly context specific.

Approach	Strengths	Weaknesses	Applicability
P2P (Custom)	Cheap and simple with two systems. Good for the pilot	Provides tight coupling with many systems. Reduces agility and flexibility with many systems	Can be used for pilot and for few systems.
H&S (Mirth)	Could provide loose coupling. Needs wider organizational collaboration.	Expensive Documentation and CDA API access. Difficult to use.	Can be used if there are enough resources and knowledge.
ESB (Mule)	Good free documentation. Easy to use. Provides loose coupling, flexibility and agility for many systems	Data Mapper crashes with CDA schema. Needs wider organizational collaboration.	Too complex for a pilot. Best for many systems.

Table 7.1: Comparing Approaches.

The H&S approach was not successful. While it was less complex than the ESB approach it turned out to be difficult to develop due to lack of extensive documentation, CDA API and a complex user interface. Compared to a P2P approach it can potentially provide loose coupling by using a CDM. This makes it a more advanced solution and it requires a higher degree of knowledge on standards and interoperability paradigms. It could be used to provide loosely coupled systems and flexibility. Each system only needs one interface to be fully interoperable with all systems in the solution. Thus, it could be used if one has enough resources and knowledge on how to use Mirth Connect.

The ESB approach is more advanced than the other. It provides a loosely coupled decentralized solution compared to an centralized Hub and Spoke. A CDM is also used in the approach making it more advanced than a P2P approach. Thus it does not fit a DHIS2 pilot compared to a P2P because it requires more time, knowledge and resources to develop.

The use of patterns described by Hohpe & Woolf (2003) for designing the second solution proved useful. The fact that Mule is based on such patterns meant that it was easy to find the correct Mule Elements. On the other hand the H&S approach , was developed without knowledge of such patterns making it difficult to grasp how translation could be executed. Other interoperability projects, using a CDM, could benefit from patterns usage at the design phase.

To summarize, all three approaches had strengths and weaknesses and could be used by other departments in Colombia in different stages depending on their resources and goal for a given interoperability solution. In addition, using a a MIF validator proved helpful because it provided explanation for errors and warnings. This is similar to the recommendation from Spronk & Grieve (2008). A MIF validator is available from the free Eclipse plugin used in the first implementation.

7.1.3 CDA as CDM

A heterogeneous landscape of systems can be integrated in an effective way if there is a canonical data model to reduce the interfaces. If there are only two intervening systems there no need for a CDM. However as more systems connect to the ESB the amount of interfaces are reduced to x for x systems. Thus a CDM is imperative for reducing the amount of interfaces. The two implementations used different CDM that provided different challenges. This is also true when the CDM is the only one using CDA.

The different CDM provided several challenges that were manifested in the planning and development of the integration solution. A common challenge was the complexity of the CDA, that makes it difficult to create a valid XML. The official HL7 Colombia implementation guide that was used to see an example CDA turned out to contain errors upon MIF validation. These errors are far from exceptional. As mentioned in chapter 2, Spronk & Grieve (2008) shows many CDA instances that contained errors upon MIF validation.

The CDA documentation is extensive and time consuming to understand. As mentioned in chapter 5 it requires a team of experts to develop an example message (Castrillón et al. 2012). While the example CDA for the First Implementation was valid, it contained warnings for the OIDs that were just example OIDs and its attempt for level three is not a good solution because the entries are just standard text. The Second Implementation used a CDA XSD and produced instances that gave errors, due to the complexity of the schema.

The choice of CDA level for the Second Implementation was changed as syntactical challenges arose. The first choice was a level one CDA instance with a non XML body, but due to challenges with the CDA schema, a structured body was created. Thus going for a level two CDA. However the implementation lacks LOINC codes for the different sections.

Dolin et al. (2006, p.38) argues that

> "many will choose to use CDA R2 in its easy-to-implement form of just a header wrapping a non-XML body or of a header with sections containing

> only narrative and no entries. This will serve to bring a log of clinical documents into a standard format. While it may be a small step, it is then possible to incrementally add structured entries".

The argument resonates for the Second Implementation CDA instance because I went for a lower level CDA that could later evolve into a level three CDA instance.

Implementing a level two CDA opens for a gradual enhancement to a more semantic model. Boone (2011, p.20) claims that "Semantic Interoperability is not a switch that you turn on and off. It's a rheostat that you turn up or down." That is why I decided to go for a shade of a low CDA level rather than to go for the most complex and highest CDA level in the Second Implementation.

Grahame Grieve, an HL7 expert, argues in a personal exchange that CDA "was intended to be a 'document' with supporting data. That's not really a suitable thing to use as a 'Canonical Data Model' – and so I don't doubt that you struggled with too much complexity." (personal correspondence, 2.6.2014). Thus the encountered CDA complexity was partly due to the fact that the CDA is not intended to be used as a CDM. An alternative the using CDA as CDM would be to develop an own data model.

However developing an own canonical data model would require much more effort and such work is out of scope of my research. If, on the other hand the SDSC IT team had developed an own data model it could be used in Mule in a similar fashion. It would have required less complexity, coding and documentation.

7.1.4 Using SOA Principles

While the research does not focus on the creation of an architecture it does use some SOA elements that are defined by Rosen et al. (2008) for the second architectural approach (chapter 6). I created a *business process* that was composed of two *integration services*. Both services were implemented as adapters. A *business service* was not created because only 2 systems were relevant. It could easily be added by adding a 'route all' Mule Component to the DHIS2 adapter. The business service would pass the CDA object further in the flow and not have any other routing logic. Its reason would be to prepare the solution for more complex interoperability.

The *document* would be the instance of the CDA. The *existing systems* were SVIGILA and DHIS2. *transformation* was done to translate from SIVIGILAS format to the CDA and from CDA to DHIS2 format. The translation also involved semantic elements. Thus both integration services *communicated* to transfer messages, using an ESB to enable *integration*.

The Second Implementation is SOA oriented rather than application oriented integration (EAI). The DHIS2 adapter uses three application API (persons, enroll, event) to create business value in one integration service. Thus exposing health event business functionality and hiding the application complexity in the REST API. It abstracts the application API and serves a business objective (Registration of a health event). The abstraction contributes to making the solution a SOA based approach.

This means that orchestration services and business services could be developed to meet business goals in the future. The solution provides a degree of loose coupling. Additionally it would give adaptability, agility and flexibility to the interoperability process.

7.2 Discussion in Terms of the Interoperability Framework

Figure 7.4 shows the interoperability framework that was presented in chapter 2 with different levels and how they were formed based on the empirical work. There was two iterations of the framework. The first one was done in the first implementation as shown in chapter 5 on page 67. The second iteration was done in the Second Implementation that is described in chapter 6 on page 79.

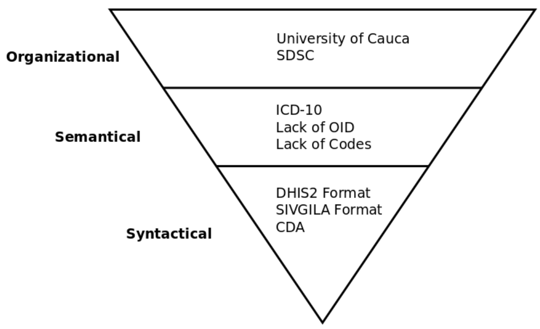


Figure 7.4: Interoperability Framework.

7.2.1 Organizational Level

Organizational interoperability is needed in any integration solution. That is: without an organizational need, any technical solution will be superfluous. A clear understanding of the needs and by extension the relevant systems that should be integrated is vital if an integration solution is to be attained.

The organizational setup, shown in chapter 4 on page 51, for developing the interoperability solution with the University of Oslo, The University of Cauca and the SDSC gave several challenges in terms of collaboration.

The SDSC had two main goals, the first one was to deploy DHIS2 as quickly as possible and the second one was to provide a general interoperability solution. A point to point solution is the fastest and easiest way to make two systems interoperable, but would provide limitations for general interoperability. Thus there were two conflicting needs. General interoperability does not only take more time to develop, but it has also higher costs and needs more experience and knowledge both for development and operations. Organizational diverging goals resulted in favoring one approach over the other: The point to point solution was thus favored. It resulted in the abandonment of the Hub and Spoke approach with Mirth Connect. Even if this solution would be technically feasible it would have been unacceptable for SDSC due to its price and complexity.

A central finding is the absence, in the project, of a Colombian National authority that has the task of defining interoperability mechanisms. The missing formats, OIDs and codes lists lead to each system having its own code lists and that the CDM could not use shared code sets. Thus making semantical loosely coupled interoperability impossible.

Colombia also lacks national guidelines for HL7 CDA. A CDA solution may encounter problems being accepted by national authorities.

There also seemed to be some doubt about what data from the 30 systems should be transferred to DHIS2. It later turned out to be used individual data from only three systems: SIVIGILA, RUAF and RIPS and aggregated data from the rest.

In conclusion there were several challenges on the organizational level that influenced the development of the interoperability solutions: Conflicting needs at the SDSC, absence of a national standard authority and HL7 guidelines. Limited planning contributed to the encountered challenges at various levels.

7.2.1.1 Different Rationalities

Chilundo & Aanestad (2005) claim that different rationalities are challenging was true for this research. The different actors had different rationalities. The University of Cauca and SDSC had conflicting instrumental rationalities. The Cauca University had a distinguished member of HL7 Colombia involved in project and a parallel research project that entailed creating the same CDA instance with the SIVIGILA form. It would be rational for the University to use CDA and gain knowledge on using the standard and Mirth Connect. The University did not take into account the price of Mirth appliances and Mirth Support prior to developing the solution.

On the other hand, the secretariat had the goal to deploy DHIS2 as quickly as possible and did not want to use a high amount of resources on an exploratory project that entailed difficulties. For them University of Cauca's view was irrational because the project would not bring any initial usefulness.

The asymmetrical power relationship between the project owners (SDSC) on one side and the two universities on the other meant that SDSC had the final say on what type of approach would be used. Their instrumental rationality had the strongest influence of the interoperability solution. This lead to discarding the First Implementation.

7.2.2 Semantical level

As mentioned in chapter 2 on page 7 semantical interoperability is important for syntacital interoperability to exist at all. Without a shared meaning of data between systems they cannot exchange information. (Rosen et al. 2008). The semantical level contained several challenges that would shape the syntacital interoperability and restrict the effectiveness of a technical solution.

The lack of national standard codes for interoperability meant that I chose to use SIVIGILA codes in the CDM. The main codes were for the following data: Event codes, organization unit ids, ICD-10 death causes, occupation, villages, municipalities, countries and ethnicity. Data dictionaries were built for the codes and used in the Second Implementation. Thus SIVIGILA became the semantic standard for the whole interoperability solution. This fact may give challenges if later added systems have information that is not covered by SIVIGILA codes. It also lead to tight semantical coupling because the Mule DHIS2 adapter has to know the SIVIGILA codes. Even though my solution provided syntactical loose coupling, a semantical loose coupling was unattainable. On the other side it was less complex to use the SIVIGILA codes for making the two systems interoperable because I only have one mapping out to DHIS2.

7.2.3 Syntactical Level

I encountered challenges mentioned by Hohpe & Woolf (2003) and Erl (2004) (chapter 2 on page 7) in relation to issues with creating interoperability for heterogeneous systems. The syntactical level in the interoperability solutions contained three main formats each with their own challenges. The first format comes from SIVIGILA. The challenge in terms to the format was that the data contained errors for organizational units. It meant that reported events with erroneous organization unit IDs would be rejected by the DHIS2 Mule Adapter. This meant that there were data quality issues that manual procedures corrected. A meaningful interoperability needs data validation at the source systems.

The second format is the DHIS2 format. The challenge found here was the fact that the documentation for the version used in this thesis changed for the API (Person to Tracked Entity) when the second Implementation was developed. It is regrettable that older versions of DHIS2 don't keep their own documentation easily available. Thus Hohpe & Woolf (2003) claims that systems change provides a challenge to interoperability was manifested in my research.

DHIS2 presented also three different formats with their own interfaces to accomplish one task: the registration of a health event. The DHIS2 adapter needs advenced transactional logic to cover this divergence between a task and its implementation.

The third format is the CDA's with two challenges related to its complex structure, its spread out and difficult to read documentation and problems with the XML schema.

As seen in chapter 6, CDA documentation was spread out and difficult to understand. I gained the experience that the CDA is so complex that even the official documentation is hard to grasp. It is either complete and difficult to read or incomplete and ambiguous in other publications.

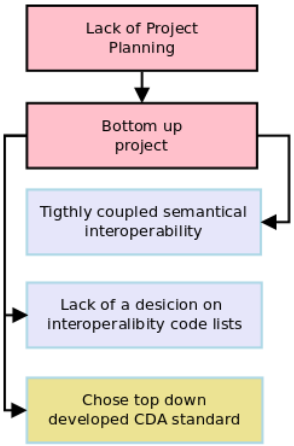
The CDA schema uses constructs that prevent automatic JAXB binding. It forces manual adaptations that are challenging to find and solve.

I had to choose the CDA level I needed. The CDA body could be either non-XML or structured. Difficulties with binding the XML schema, as shown in chapter 6 on page 79 meant that it was initially not possible to use a non-XMLBody. Thus the complexity was partly shaped by this technical problem: It resulted in a structured body. The non-XML body would have produced a simpler solution.

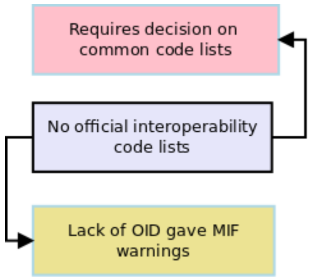
Another syntacital challenge was manifested in the time and data format CDA uses. According to Boone (2011) it uses ISO 8601 standard for time and date meaning that all the dates and time should be translated from SIVIGILA format to the ISO standard for a valid CDA. It was not done in the implementation but illustrates the complexity of CDA and how it would mean more effort for development and test.

I also chose codes for confidentiality without having the authority to do so. These fields are mandatory in the CDA. Thus creating a syntactical valid CDA requires an authoritative organization that takes these decisions.

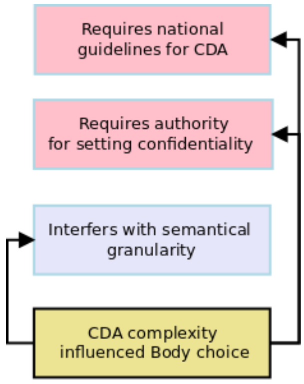
There are also different ways of showing missing values (Boone 2011). It is done by choosing between many null flavors. Each of them codes the absence of information. Deciding which null flavour to use in each case requires the participation of the organizations running the source systems and their understanding of the lack of information. Thus using CDA is a organizational demanding solution.



(a) Organizational Level Influences.



(b) Semantical Level Influences.



(c) Syntactical Level Influences.

FIGURE 7.5: Various Interoperability Levels and their influences. (Colours: Pink = organizational level. Purple = semantical level. Yellow = syntacital level).

7.2.4 Interaction between the Levels

The use of the CDA standard and a bottom up project entailed a mixed standardization strategy. It has similarities to the anticipatory standardization strategy and a more bottom up project strategy meaning that does not fit in the categories described by Hanseth et al. (2012).

A top down developed standard was used. The CDA standard itself was designed by HL7 to fit anticipated needs. Thus it forced a top-down element to the project. This entailed that the installed base was not used to its fullest potential. Rather than drawing on the installed base of the 30 systems to create an own CDM to fit their need based on working solutions, the CDA was chosen. Mainly because there was not enough time and resources to develop a bottom up CDM. An endogenous bricolage process could be used to create an own CDM by using the resources available: the processes of 30 systems. This would draw upon the installed base and possibly create a less complex CDM tailored to Caucas needs.

At the same time the project followed a bottom up approach. Both implementations were developed without guidelines from a national level. They would solve Caucas interoperability needs without taking into account different solutions that can emerge in other Colombian departments. Thus a decision at a national level to use a different standard than the CDA would entail for Cauca to change the solution's CDM. The combination of bottom up and top down strategies influenced the different levels. Figure 7.5 shows how the levels influenced each other.

Figure 7.5(a) shows the how the organizational bottom up strategy was used to chose a top down developed standard (syntactical). Furthermore it was done without national guidelines and a strategy for common shared semantical code lists. It lead to tight semantical coupling in the interoperability solution. The bottom up project was also influenced by a lack of planning: The cost of Mirth appliances was not taken into account before starting the development.

The semantical level influenced the organizational and syntactical level as shown in figure 7.5(b). No organizational agreement on defined semantical interoperability code lists meant that I had to choose interoperability data dictionaries from SIVIGILA. It made DHIS2 and SIVIGILA tightly coupled at a semantical level: If SIVIGLA changes its semantical codes it will affect the DHIS2 adapter. Additionally we lacked OIDs as Colombia only has a country OID. The syntactical level was influenced by the lack of official OIDs and gave warnings when using a MIF validator.

The syntactical level (figure 7.5(c)) required and influenced the semantical and organizational levels. The CDA is written in a complex way leading to problems using JAXB bindings. It influenced in the choice of CDA body: from a planned unstructured body to a structured body. A structured body has more options for semantical codes for each section and codes for potential level three entries. On the other hand an unstructured body has no sections and no entries meaning semantical granularity is reduced. Additionally, the CDA required defining confidentiality. This should be decided at the organizational level. Working with the CDA at the syntactical level requires organizational decisions and authorities.

To summarize, the mixed process shows that interoperability for Cauca Information Infrastructure, is more than a purely technical solution because it is strongly influenced by organizational, syntactical and semantical dimensions.

Choosing CDA as the CDM required an organizational maturity that Colombia (and therefore Cauca) was far away from having achieved. This means that in a successful project there must be a correspondence between the syntactical and the organizational levels.

On the other side, following a bottom-up project tried to alleviate the absence of semantical and organizational maturity and produced a sub-optimal solution.

In addition, if other departments use their own CDM that is different for other general interoperability solution it could cause problems if information from one department is to be transferred to another. Thus there is a need for deciding on the use of standards across departments from a national level to reduce future complexity.

7.3 Connecting Elements with Gateways

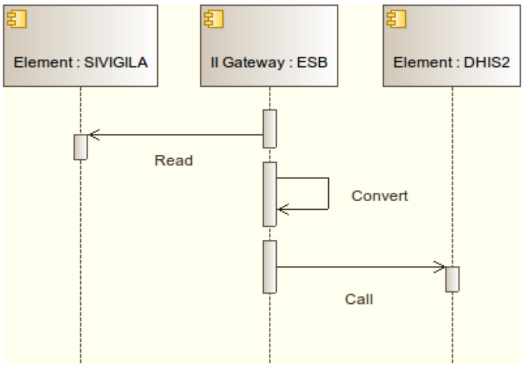


FIGURE 7.6: Gateway for DHIS2 and SIVIGILA.

A gateway (Hanseth 2001) was used to link elements. By simplifying the technical process of the interoperability solutions they can be seen as a gateways linking two elements (DHIS2 and SIVIGILA). I will here discuss the Second Implementation as a gateway. The Hub and Spoke approach did not successfully link the systems and will not be discussed.

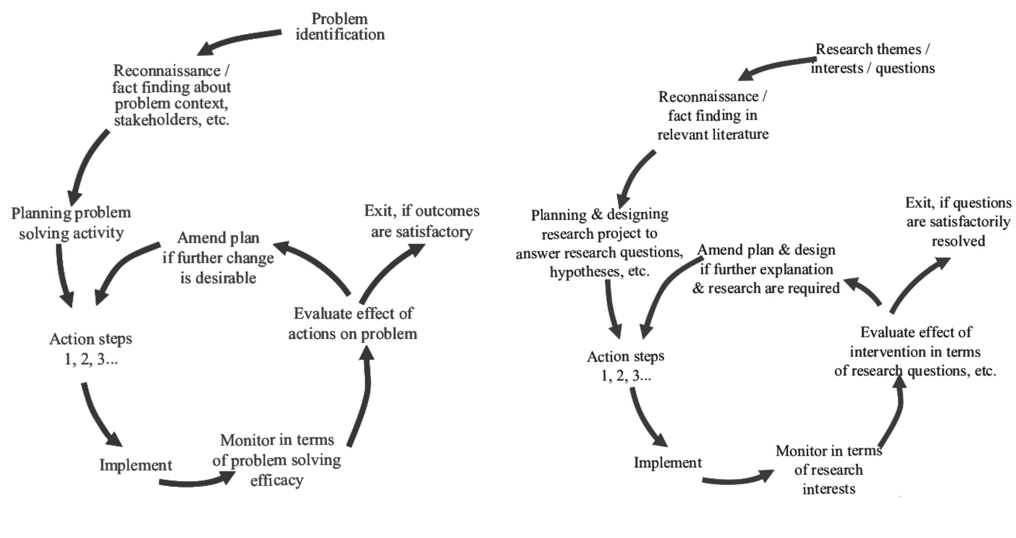
Figure 7.6 shows the gateway that was used in the second implementation[3] . The gateway reads data from an element (SIVIGILA), converts data and transfers it to another element (DHIS2). Thus there is no need to have the different elements use the same standard if they use a gateway standard (CDA). The created gateway makes the elements loosely coupled. The point to point approach can also be seen as a gateway but providing tightly coupled elements.

As shown in section 7.1 the type of solution forms the evolution of Caucas II. Thus a gateway that provides tightly coupled elements creates inert evolution compared to an more loosely coupled active evolution. This in turn impacts Cacuas II in terms of adaptiveness and flexibility: changing, adding and removing systems. However, a gateway without properly defined processes might implode.

7.4 On research

This research covered two AR cycles. Each of them with different challenges. As stated in chapter 3 the first cycle was plagued with little research and match action. The issue lead to executing a second cycle. This section shows how McKay & Marshall (2007) concept of dual imperatives influenced the AR cycles and why I chose to do two cycles.

[3]For details on the ESB see chapter 6 figure 6.6 on page 86.



(a) Action Cycle.   (b) Research Cycle.

Figure 7.7: Action and Research Cycle within one Action Research Cycle (McKay & Marshall 2007)

7.7 shows the action cycle (7.7(a)) and the research cycle (7.7(b)) that together forms the dual imperatives of AR[4].

The first AR cycle lacked knowledge on research in general. It influenced the research cycle because the initial research question was not based on knowledge on the topic: There was no time for fact finding in the literature. Designing the project to answer the research questions was not done because I did not know the concept of research design. Furthermore, monitoring in terms of research interests was also not focused on due to the lack of knowledge on research. It also influenced the lack of evaluation of the effect regarding actions in terms of research questions.

On the other hand the action cycle of the first AR cycle was carried out differently. The problem identification was accomplished. Fact finding about the problem context was done by collecting documents and interviews. The problem solving activity entailed developing the First Implementation. Monitoring problem solving efficiency was also done with the SDSC IT-team. Evaluation of the action was discussed with the IT-team. However I did not base my action on technical oriented literature. This important aspect is missing in McKay & Marshall (2007) model for action. That is, action requires knowledge that can be found, in my case, in the practitioners literature.

The results of the first AR cycle was more practical knowledge than research knowledge due the cycle being more action than research oriented. Consequently, I decided to conduct a second cycle when I had more knowledge on the topic and on research in

[4]See chapter 3 figure 3.2 for the full figure and details.

general. The input to the second AR cycle was mostly practical knowledge generated from the first cycle but also the experienced challenges for conducting AR.

The second AR cycle was more balanced in terms of action and research. The research cycle was influenced by properly defined research questions and I found and explored technical and academic literature on the topic. I also planned to answer them by designing a research process. I monitored and evaluated in terms of answering the research questions. It was done after and during action taking.

The action cycle was influenced by knowledge on how to solve the problem found in the first AR cycle. I planned and designed the how to solve the problem with integration patterns and by communicating with the Cauca IT-team. Actions were taken to solve the problem. Monitoring and evaluating the actions was done by myself and by discussing it with the IT-team.

The second cycle had more satisfactory outcomes in terms of the research cycle and the action cycle. The AR was stopped based on the outcomes and lack of time. In addition to the dual imperatives, I met other challenges in the AR cycles. The following section describes the challenges in terms of three threats to AR.

7.4.1 Three Challenges

Both AR Cycles faced issues presented in the literature: The dual imperative and the three threats to AR.

In terms of the dual imperative, the first cycle was action focused rather than balanced. Lack of knowledge on AR, HIS and general research resulted in over focusing on problem solving rather than a balanced dual imperative. On the other hand, the second cycle was carried out with the knowledge that was lacking in the first cycle. However my work was performed in Norway meaning that communication with the IT team was reduced to Skype calls, chat and emails. Thus conducting AR with a geographical distributed team meant that both actions and research processes were delayed due to misunderstandings and lack of direct access to the team.

Kock (2007) AR threats described in chapter 3 were also encountered. The *subjectivity threat* was manifested in the second cycle because the design of the solution began before any consultation with Cauca. After experiencing being involved with the SDSC IT team and the failure of the first cycle I was also sceptical towards their technical reasoning regarding the CDA.

The *uncontrollability threat* was manifested, in the first cycle, due to the lack of control over the chosen interoperability software and over the health standard. The lack of funds influenced and restricted the problem solving due to the cost of documentation for MIRTH and access to the CDA API. It also manifested itself in the second cycle because of the long periods of waiting time caused by communication issues.

The *contingency threat* was encountered because the solutions were context bound and specific to Cauca in Colombia. The data from the interviews and communication was also context bound. Thus the difficulty to generalize the findings.

7.4.2 Fluid AR borders

As mentioned in chapter 3 on page 35 Bjørn & Balka (2009) claim that AR borders are fluid. The 'fluid' borders of the AR phases in both cycles were manifested differently.

The first cycle had deadlines and collaboration with the SDSC IT team had to be planned due to the busy schedule of the participants. Thus control over the phases and borders were more rigid in comparison with the second cycle. The second cycle lacked hard deadlines and the technical solution was developed without direct participation of the IT team in Cauca. However lack of communication meant that the fluid borders between the phases were still dependent on collaboration.

The planning phase for the second cycle had no clear time borders. Evaluation was done as actions were performed rather than following a strict plan. It shows that my second AR borders were more emergent and fluid than the first. The external control that influenced the phases was present in both cycles but distinct.

Thus working in the SDSC IT team imposed more strict borders while working more isolated gave more fluid borders. A direct result of this in the second cycle was that the planning phase had less importance than in the first cycle.

7.5 Summary

This chapter presented different architectural approaches, organizational, semantical and syntactical challenges generated by both implementations. The Hub and Spoke approach with mirth connect proved a failure but could potentially reduce interfaces to x interfaces per added system. On the other hand the ESB approach with Mule was a success as a proof of concept and provided a degree of loose coupling. The approaches lacked institutional involvement for the use of HL7 CDA at a National level and semantical codes for interoperability. The three different levels for the framework of interoperability influenced each other for better or worse.

Chapter 8

Conclusion

8.1 Concluding Remarks

This thesis set out to explore the process of creating interoperability for HIS. It generated different challenges facing the process of making SIVIGILA and DHIS2 interoperable in Cauca, Colombia.

The practical research work included two AR cycles with two different architectural approaches. The First Implementation was unsuccessful: The SCDC goal of rapid implementation forced the adoption of point to point integration with lower interoperability and tight coupling. Tool and solution deficiencies resulted in a technical unfeasible and and organizational/financial nonviable solution that was abandoned.

On the other hand, the Second Implementation was successful as a proof of concept. As the first implementation, it used a CDA as its CDM, but this time using the full standard XSD instead of an example XML. Part of its results was disseminated to the IT team in SDSC and they began using it. This implementation created knowledge on different architectures, approaches and tools. It applied practitioners and academic literature.

The research questions presented in chapter 1 can be answered based on the results of the AR cycles. The main research question was:

What are the challenges of making DHIS2 and SIVIGILA interoperable using HL7 CDA, in a landscape of distributed heterogeneous systems in Cauca, Colombia?

The encountered challenges were organizational, semantical, syntactical and technical. They influenced the architectural approach for both interoperability implementations. At the organizational level, it was shown that this interoperability solution requires a maturity capable of providing a shared knowledge process, an organizational structure and a recognized authority. At the semantical level, it needs knowledge (code lists) and structure (OIDs). At the syntactical level it requires a shared format on which to build the solution.

The answer to the sub research questions are:

- *How does the architectural approach influence the interoperability solution?*

  There were several ways the architectural approach influenced the solutions. Firstly, a Point to Point gave close coupling and hampered agility and flexibility if expanded. Secondly a Hub and Spoke approach proved difficult due to the chosen tool and CDM has inherent limitations due to centralization. Thirdly, the ESB approach proved successful as a proof of concept but had limitations due to organizational issues.

- *How can semantical interoperability be achieved between the two systems?*

  Semantical interoperability was limited in the First Implementation. It did not use any official interoperability code. It resulted in the researcher choosing a limited set of data dictionaries. The second approach achieved only a tightly coupled semantical interoperability. due to the absence of shared knowledge in the form of official codes and OIDs.

  Semantical interoperability is higher for complex data models like the CDA. Semantical interoperability can be achieved either by simpler data models or a higher shared knowledge. It is important to find the right balance between organizational maturity and the data model.

- *What are the syntactical and technical challenges to make the two systems interoperable?*

  The syntactical challenges related to the CDAs complexity, SIVIGILAS data errors and DHIS2 documentation. The complexity of CDA was aggravated by problems of spread out documentation and the readability of the standard CDA documentation. The CDA schema itself produced difficulties with JAXB binding. The quality of input data from SIVIGILA was low presenting difficulties that were not solved. The DHIS2 documentation changed inadvertently during the research.

  Syntactical and technical interoperability needs data models of acceptable complexity and technical appropriateness as well as stable and well documented systems.

- *How can organizational factors influence the interoperability solution?*

  Organizational factors such as power asymmetries, different rationalities and divergent goals shaped the choice of the interoperability solutions. Lack of national guidelines and official interoperability codes influenced interoperability producing semantical tight coupling. The absence of a national or regional authority determining standards for interoperability and rules for information forced the researcher to decide parts of the standard instances, codes and access rights.

  The presence of a recognized authority defining an interoperability process is crucial for wider integration across Cauca. An appropriate authority for making standards and guidelines on HIS interoperability should be involved in the process of making systems interoperable..

- *What strategy can be used to make them interoperable?*

  The strategy that was used contained elements of top down and bottom up approaches. It was bottom up because it met Cauca's needs without being based on national guidelines nor taking into account solutions for other departments. On the other hand it used a standard that was developed top down and forced such an approach.

  The bottom up approach did not have official backing for the decisions taken due to the absence of an interoperability authority. The top down approach meant that the chosen standard was not tailored to Cauca's needs, adding a complexity that the Ministry of Health did not approve.

  Central financial and operational aspects were not resolved prior to undertake the development/research producing solutions that even if technical successful would be unfeasible.

> If it was to be done again the strategy could benefit from proper planning with cost and calculate return of investment[1]. In addition an standard setting authority for Colombia should be involved in the strategy making it more top down.

The process of making SIVIGILA and DHIS2 interoperable is complex and can conceptually be categorized by the interoperability framework presented by Braa & Sahay (2012). It contains an organizational, a semantical and a syntactical level. My research shows that these levels influence each other and contain elements that could enable or restrict the interoperability solution, becoming interacting sub processes.

The main finding of my research is the importance of the organizational sub process and the need of a recognized authority defining interoperability rules, a shared knowledge platform and project strategies. Thus the organizational sub process defines how the other sub processes will evolve.

Another central finding is the need to adapt the data model, an important interoperability artifact, to the organizational maturity so that the organization can support the solution, and the solution can answer the organization's needs. This shows how tight the syntactical sub process and the organization depend on each other.

Making the information from two (or more) independent systems understandable as an integrated whole depends on shared knowledge and an organization that can define it.

The strategy for a project involving two or more systems needs a more top down approach that can back up the decisions and define a financial frame. The organizational sub process is crucial for a successful project. Additionally, a bottom up standard based on the installed base could provide less complexity than top down developed standard.

In addition to answer the research questions, this thesis delivered an empirical proof of concept for an interoperability solution involving two systems. The solution follows a service oriented architecture. It uses a CDM and ESB tools so it is suitable to support general interoperability between the health IT systems in Cauca, Colombia. The solution was formed by the constrains uncovered in this thesis.

8.2 Future Work

Research could be conducted to define a data model or try other health standards to better adapt the data model an the organization. Further work on deployment, security and other operational areas could enrich a more complete HIS interoperability approach.

[1]For information on Return of Investments (ROI) see Manouvrier & Ménard (2008).

Further research on how organizational decisions influence interoperability in Cauca could also be relevant for future work.

Appendix A

Appendix: CDA Instance for the First Implementation

The data in the figure does not correspond to a real patient. A CDA XSL style sheet was used to render the figure in the appendix. The style sheet was found in Lantana groups website (*Free Tools* n.d.) and was not used in the interoperability solution but only to view the CDA while developing it.

Datos Basicos

Patient	Adrian Arevalo		
Date of birth	December 12, 1203	**Sex**	Male
Contact info	Pirmary Home: Carrera 29A1 No. 12B-105 Telecom information not available	**Patient IDs**	55140250 2.16.840.1.113883.19.5
Document Id	c266 2.16.840.1.113883.19.4		
Document Created:	February 3, 2013, 00:15:00		
Author	Augousto Gonzalez, MD, Organization Name		
Information recipient:	Secretaria de Salud		
Document maintained by	890.307.200-5 2.16.840.1.113883.19.5		

Table of Contents

- INFORMACIÓN GENERAL
- IDENTIFICACIÓN DEL PACIENTE
- NOTIFICACIÓN
- ESPACIO EXCLUSIVO PARA USO DE LOS ENTES TERRITORIALES - AJUSTES

INFORMACIÓN GENERAL

- Codigo de la UPDG: 1900108159001
- Razon social de la unidad primaria generadora del data (UPGD): IPS nueva Popayan
- Nombre del evento: Tifus Epidémico trasmitido por piojos
- Codigo del evento: 127
- Fecha de Notificacion : 2012/07/07

IDENTIFICACIÓN DEL PACIENTE

- Tipo de documeto de indentificacion : MS
- Numero de identificacion : 94489321
- Primer Nombre : adrian
- Segundo Nomber: manuel
- Primer apellido: arevalo
- Segundo apellido: Soria
- Telefono : 9872343134
- Fecha de nacimiento :05041983
- Edad: 28
- Unidad de medida de la edad: 1
- Sexo: Masculino
- Pais de procedencia del caso : Colombia
- Pais de procedencia del caso Codigo : 1234
- Departamento/municipio occurencia del caso : Popayan
- Departamento/municipio occurencia del caso Codigo : 4567
- Area ocurrencia del caso: 1
- Localidad de occurencia del caso:
- Barrio occurencia: Esmeralda
- Cabecera municipal/Centro poblado/Rural disperso:
- Vereda: Vereda de Torres
- Ocupacion del paciente: plomero
- Ocupacion del paciente Codigo: 1345
- Tipo de regimen en salud : 1
- Nombre de la administradora de servicios de salud: SaludCoop
- Nombre de la administradora de servicios de salud Codigo: 3465
- Pertenencia etnica: 1
- Groupo poblacional: 5

NOTIFICACIÓN

- Codigo del municipio: 455
- Departamento y municipio de residencicia del paciente: Popayan
- Direction de residencia:"VEREDA EL CABILDO "
- Fecha de consulta: 2012/07/07
- Inicio de sintomas: 2012/07/05
- Clasificacion inicial de case: Sospechoso
- Hospitalizado: 1
- Fecha de hospitalizacion:2012/07/06
- Condición final: 1
- Fecha de defunción: 2012/07/07
- No. certificado defunción: 6
- Causa básica de muerte
- Causa básica de muerte Codigo
- Nombre del profesional que diligenció la ficha: Carlos Gonsalez
- Teléfono del profesional que diligenció la ficha:31040619765

ESPACIO EXCLUSIVO PARA USO DE LOS ENTES TERRITORIALES - AJUSTES

- Seguimiento y clasificación final del caso:
- Fecha de ajuste:

Figure A.1: CDA View.

Appendix B

Appendix: CDA Derived Relevant Java Classes

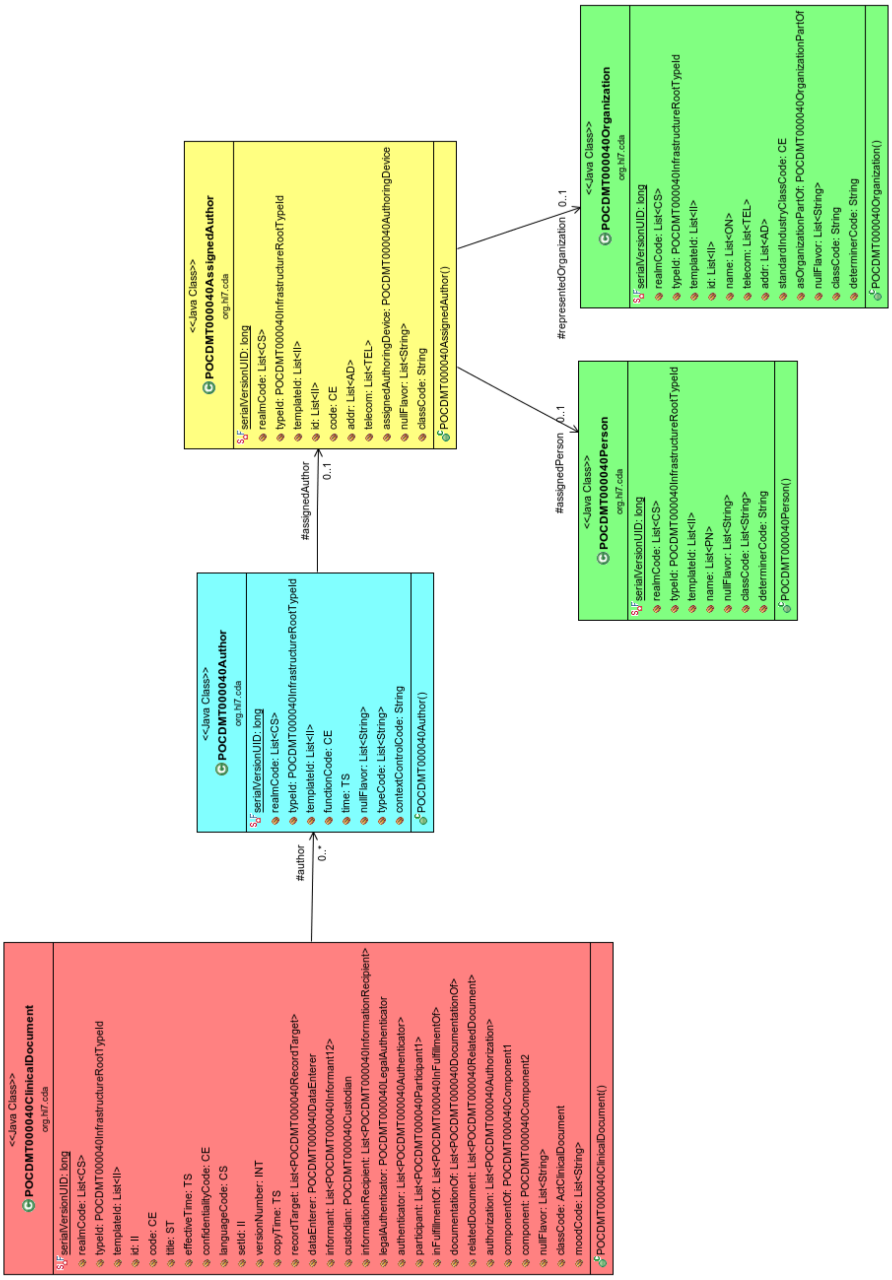


Figure B.1: Author POJO Class Diagram.

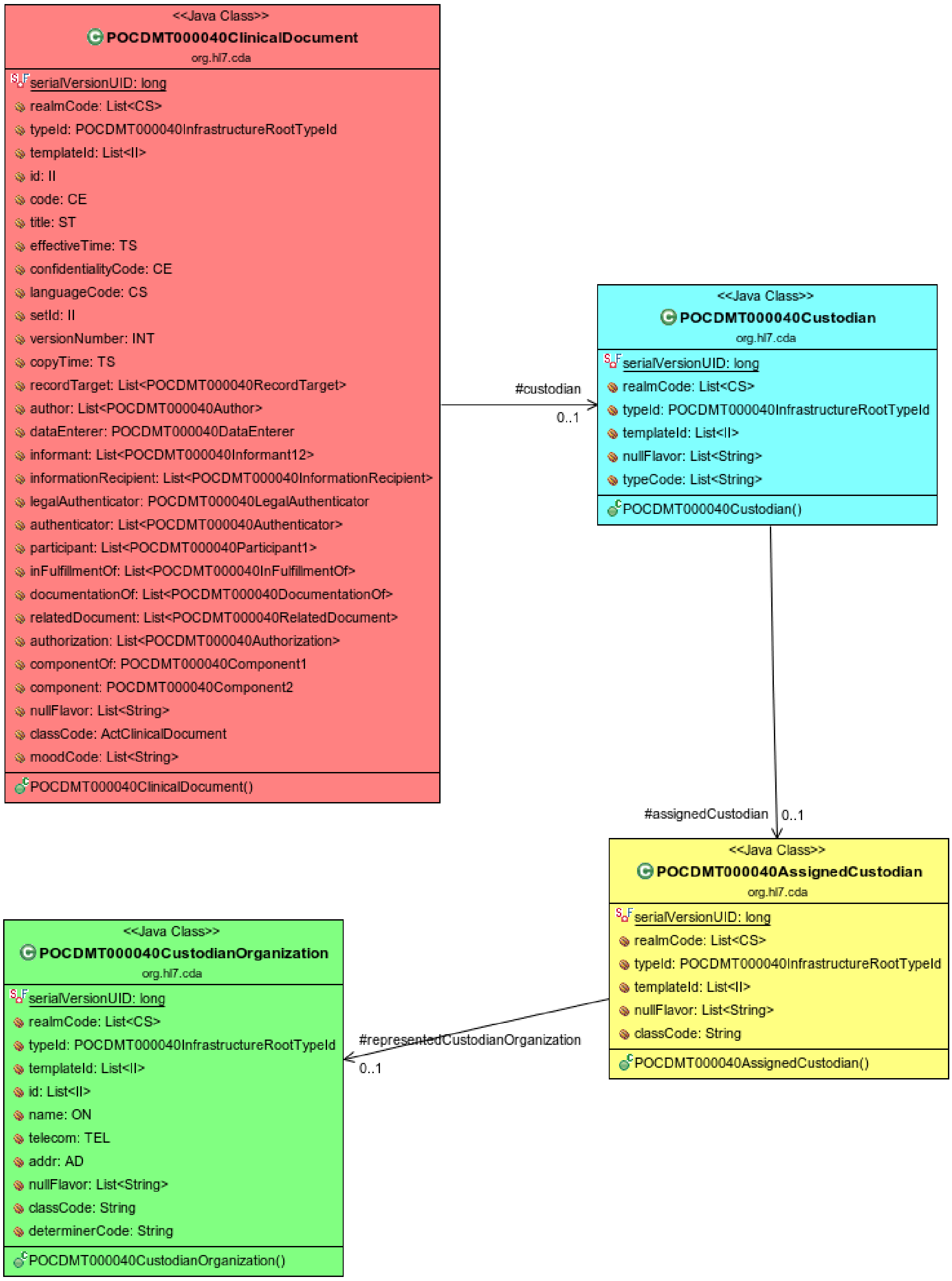


FIGURE B.2: Custodian POJO Class Diagram.

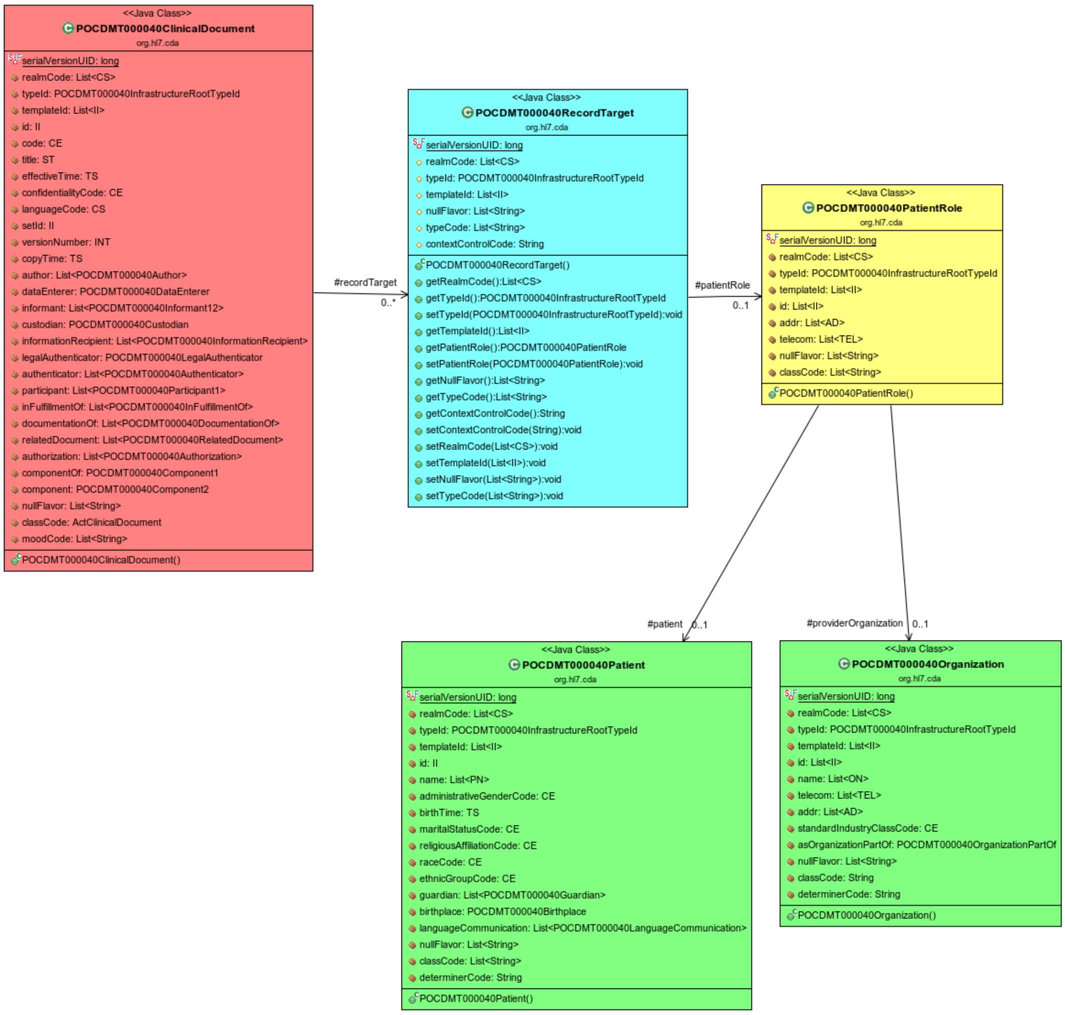


FIGURE B.3: Record Target POJO Class Diagram.

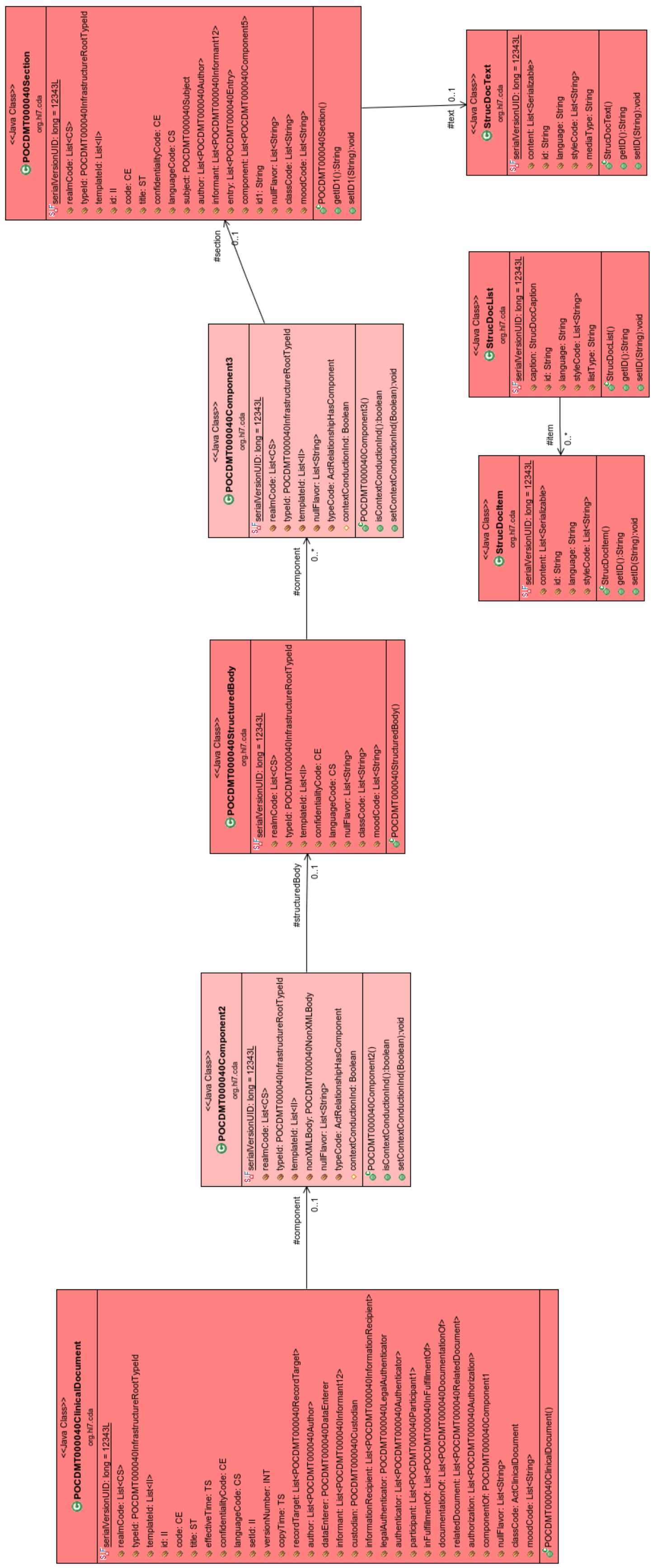


FIGURE B.4: Structured Body POJO Class Diagram.

Appendix C

Appendix: Transferring Patient Information to the CDM

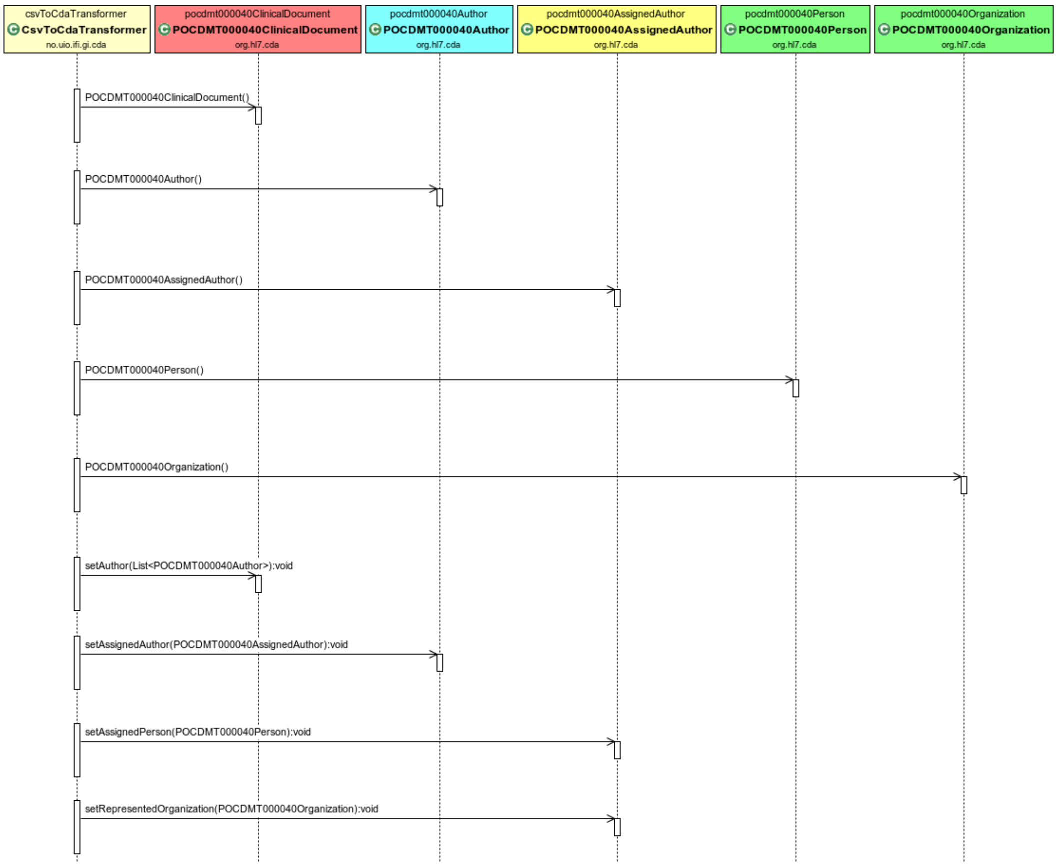


FIGURE C.1: Set Author Sequence Diagram.

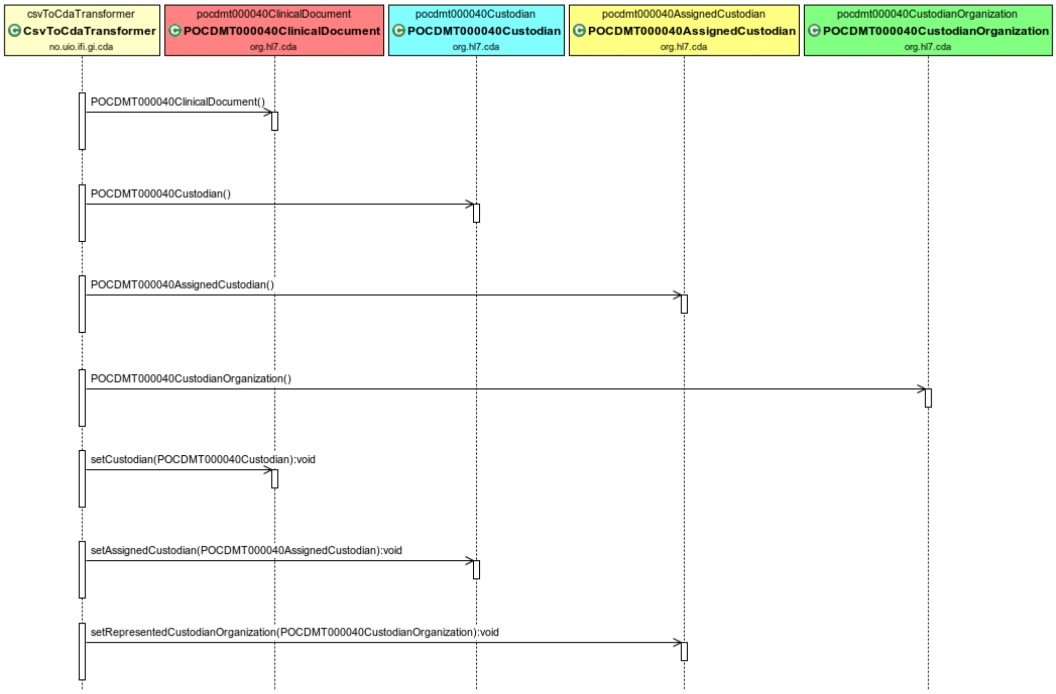


FIGURE C.2: Set Custodian Sequence Diagram.

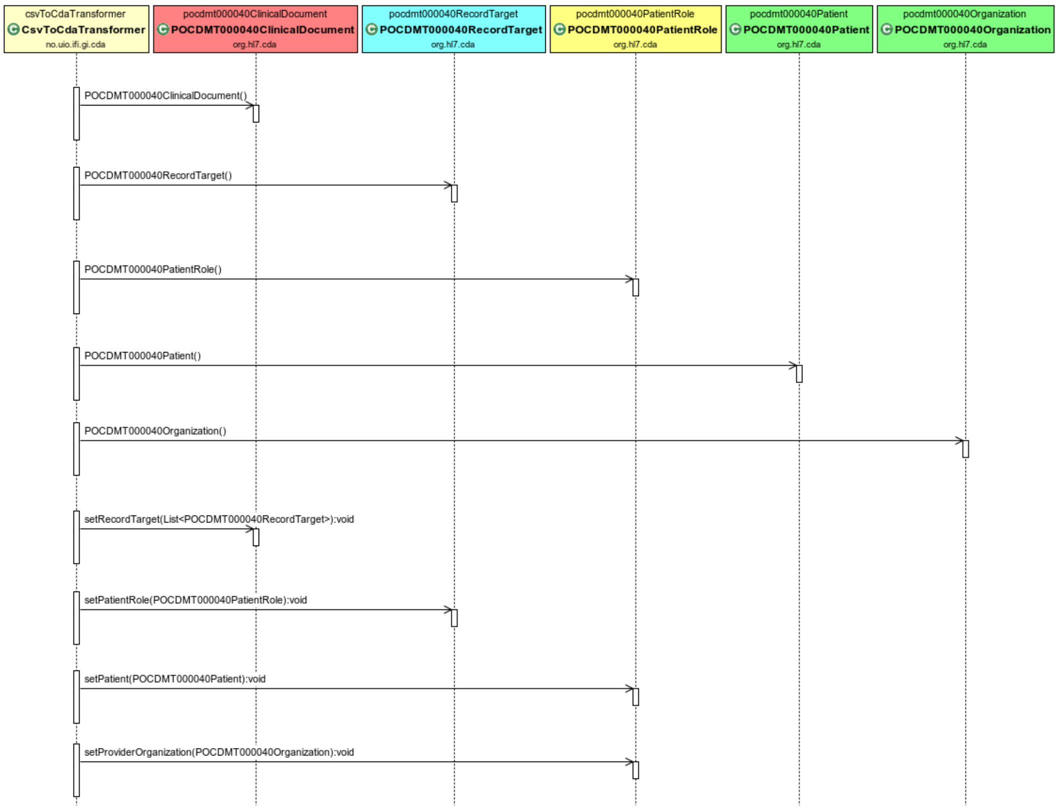


FIGURE C.3: Set Record Target Sequence Diagram.

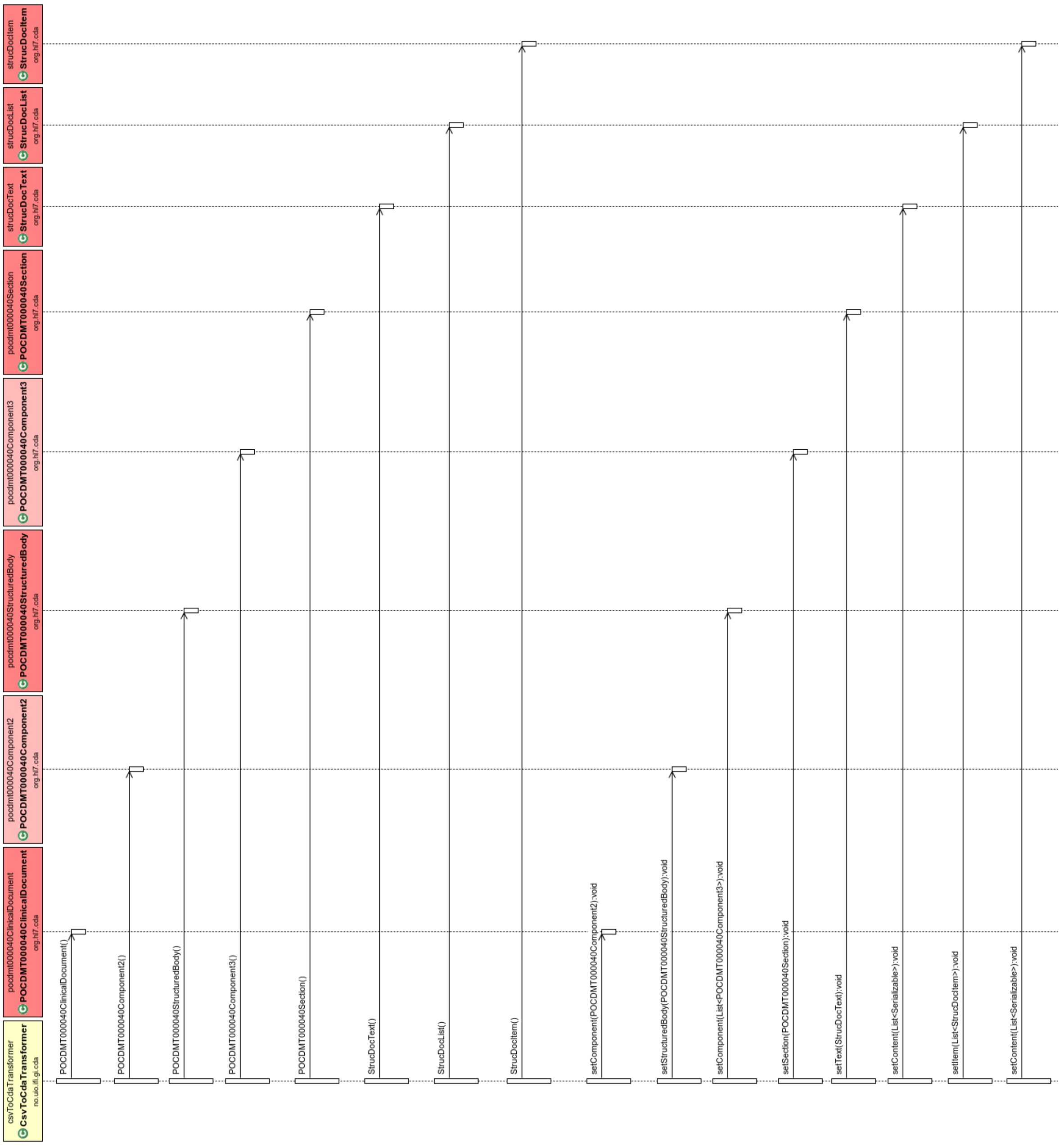


FIGURE C.4: Set Structured Body Sequence Diagram.

Appendix D

Appendix: Transferring Patient Information from the CDM

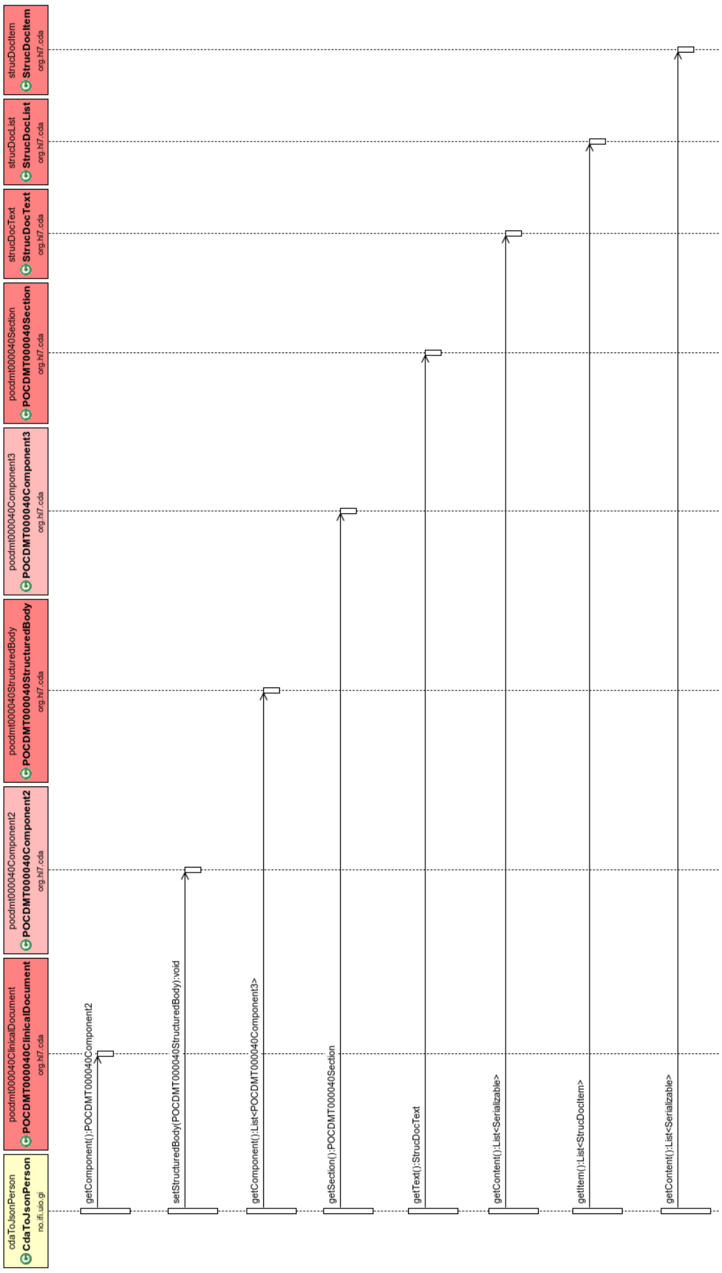


Figure D.1: Get Body for Person Translation.

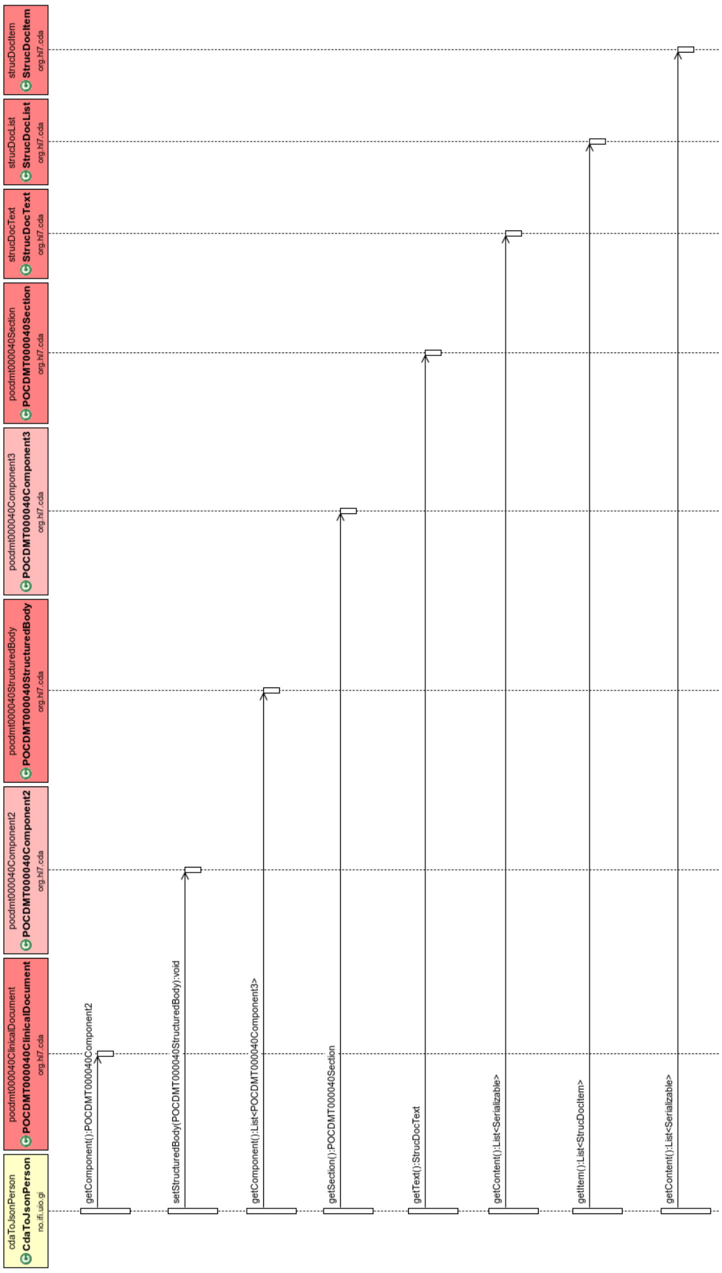


Figure D.2: Get Body from Event Translation.

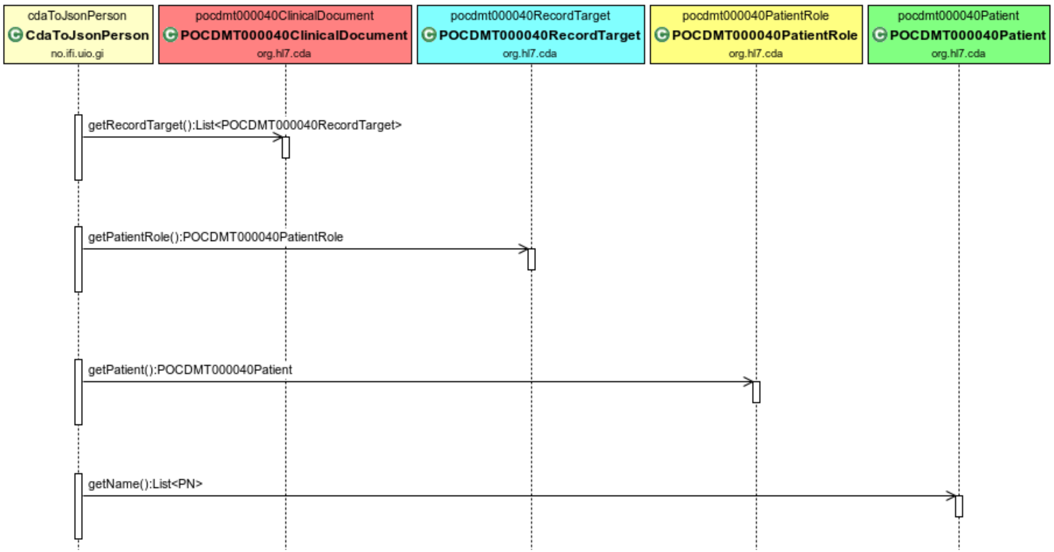


FIGURE D.3: Get Record Target for Person Translation.

Appendix E

Appendix: SIVIGILA Database Model

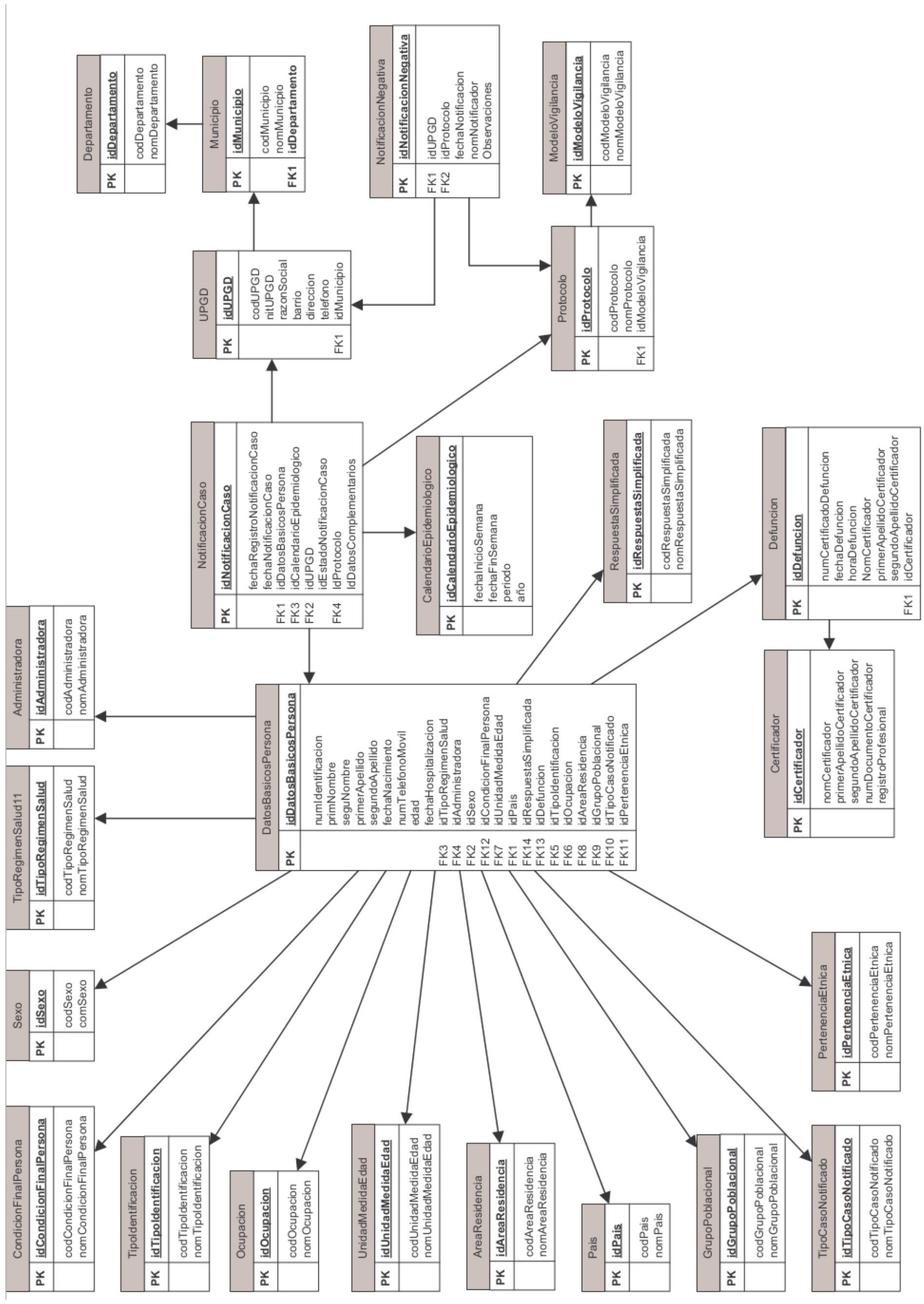


FIGURE E.1: SIVIGILA database entity relation model (Bolaños & Muñoz 2009).

Appendix F

Appendix: HL7 RIM and CDA Constrained Information Models

Relevant HL7 CDA constrained models used in the work.

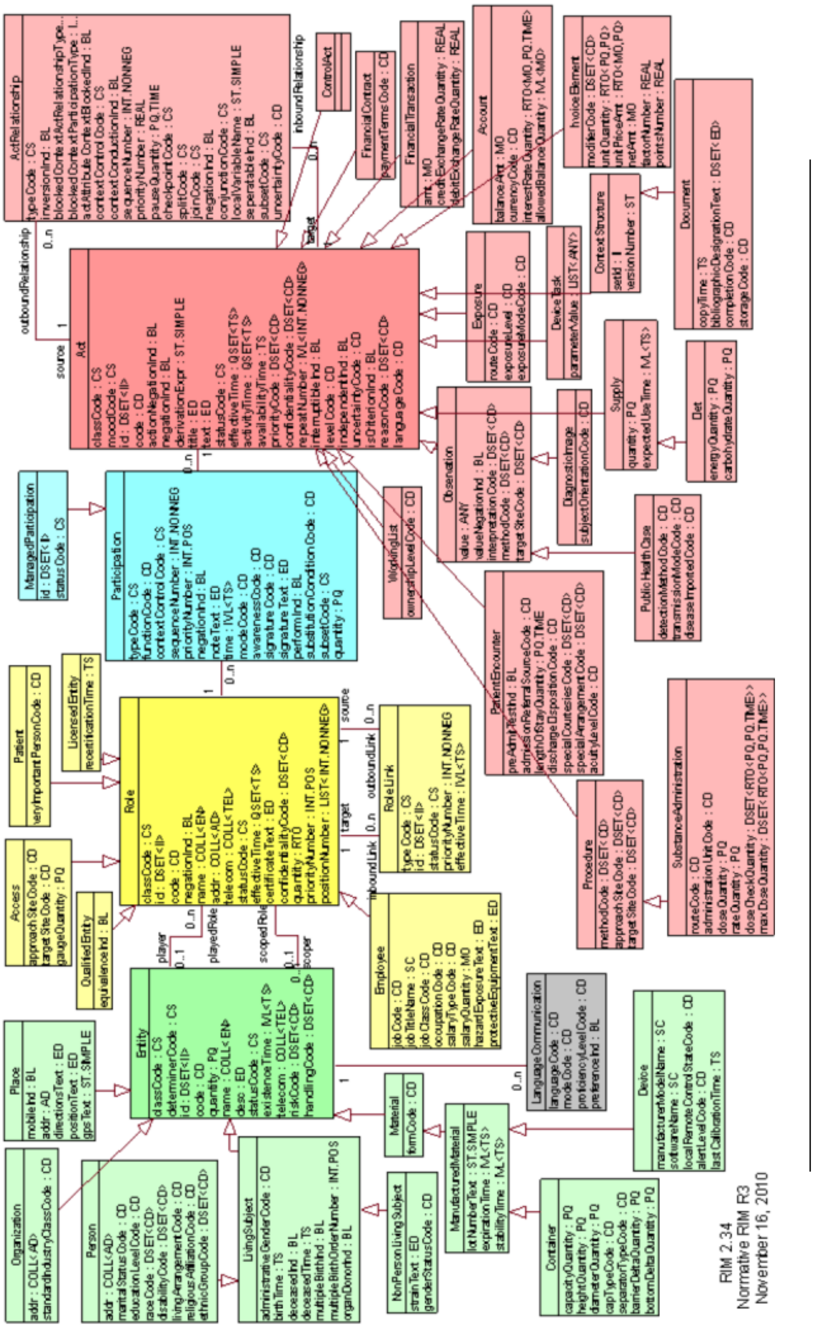


FIGURE F.1: RIM

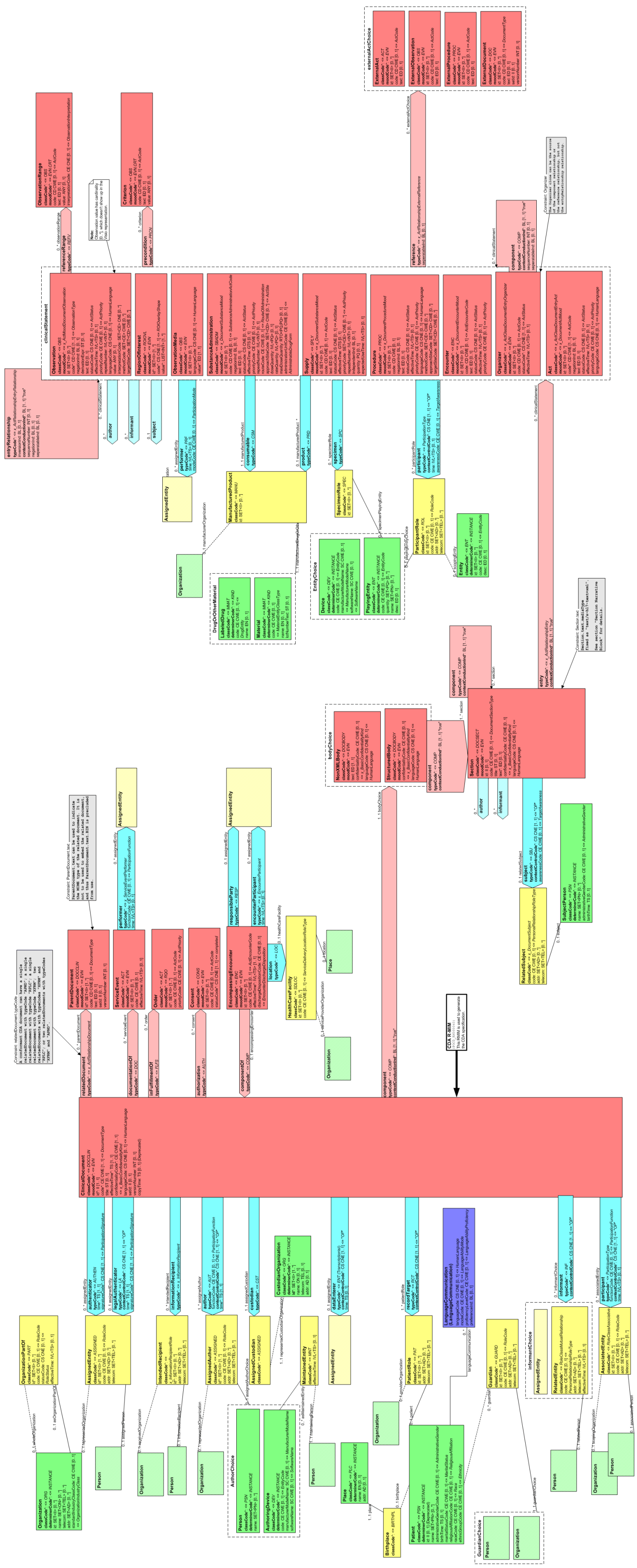


Figure F.2: CDA R-MIM

No	Element Name	Card	Ma	Co	Rim Source	of Message Element Type	St	Domain	CS	Abst	Nt
	CDA (POCD_HD000040) Hierarchical Description										
	ClinicalDocument	0..1			Document	ClinicalDocument	N				
1	typeId	1..1	M	R	InfrastructureRoot	II	D				Default: @root="2.16.840.1.113883.1.3"; @extension="POCD_HD000040"
2	classCode	1..1	M	R	Act	CS	D	DOCCLIN	CNE		Default: DOCCLIN
3	moodCode	1..1	M	R	Act	CS	D	EVN	CNE		Default: EVN
4	id	1..1		R	Act	II	D				
5	code	1..1		R	Act	CE	D	DocumentType	CWE		
6	title	0..1			Act	ST	D				
7	effectiveTime	1..1		R	Act	TS	D				
8	confidentialityCode	1..1		R	Act	CE	D	x_BasicConfidentialityKind	CWE		
9	languageCode	0..1			Act	CS	D	HumanLanguage	CNE		Default: @codeSystem="2.16.840.1.113883.6.121"
10	setId	0..1			ContextStructure	II	D				
11	versionNumber	0..1			ContextStructure	INT	D				
12	copyTime	0..1			Document	TS	D				DesignNote: Deprecated
13	**recordTarget**	1..*			Act	SET<RecordTarget>	N				
14	typeCode	1..1	M	R	Participation	CS	D	RCT	CNE		Default: RCT
15	contextControlCode	1..1	M	R	Participation	CS	D	OP	CNE		Default: OP
16	**patientRole**	1..1			Participation	PatientRole	N				
17	classCode	1..1	M	R	Role	CS	D	PAT	CNE		Default: PAT
18	id	1..*			Role	SET<II>	D				
19	addr	0..*			Role	SET<AD>	D				
20	telecom	0..*			Role	SET<TEL>	D				
21	**patient**	0..1			Role	Patient	N				
22	classCode	1..1	M	R	Entity	CS	D	PSN	CNE		Default: PSN
23	determinerCode	1..1	M	R	Entity	CS	D	INSTANCE	CNE		Default: INSTANCE
24	id	0..1			Entity	II	D				DesignNote: Deprecated
25	name	0..*			Entity	SET<PN>	D				
26	administrativeGenderCode	0..1			LivingSubject	CE	D	AdministrativeGender	CWE		
27	birthTime	0..1			LivingSubject	TS	D				
28	maritalStatusCode	0..1			Person	CE	D	MaritalStatus	CWE		
29	religiousAffiliationCode	0..1			Person	CE	D	ReligiousAffiliation	CWE		
30	raceCode	0..1			Person	CE	D	Race	CWE		
31	ethnicGroupCode	0..1			Person	CE	D	Ethnicity	CWE		
32	**guardian**	0..*			Entity	SET<Guardian>	N				
33	classCode	1..1	M	R	Role	CS	D	GUARD	CNE		Default: GUARD
34	id	0..*			Role	SET<II>	D				
35	code	0..1			Role	CE	D	RoleCode	CWE		
36	addr	0..*			Role	SET<AD>	D				
37	telecom	0..*			Role	SET<TEL>	D				
38	**guardianChoice**	1..1			Role	Person | Organization	N			Y	
39	**guardianPerson**	1..1			LivingSubject	Person	N				
40	classCode	1..1	M	R	Entity	CS	D	PSN	CNE		Default: PSN
41	determinerCode	1..1	M	R	Entity	CS	D	INSTANCE	CNE		Default: INSTANCE
42	name	0..*			Entity	SET<PN>	D				

FIGURE F.3: Screenshot of a part of the CDA HMD using LibreOffice.

Bibliography

About DANE (2013), http://www.dane.gov.co/en/index.php?option=com_content&view=article&id=175&Itemid=28. [Online], accessed 24.05.2013.

Abouzahr, C. & Boerma, T. (2005), 'Health information systems : the foundations of public health', *Bulletin of the World Health Organization* **83**(8), 578–583.

Adaletey, D. L., Poppe, O. & Braa, J. (2013), Cloud computing for development—improving the health information system in ghana, *in* 'IST-Africa Conference and Exhibition (IST-Africa)', IEEE, pp. 1–9.

Adelman, C. (1993), 'Kurt lewin and the origins of action research', *Educational Action Research* **1**(1), 7–24.

Avison, D. (2002), Action research: a research approach for cooperative work, *in* 'The 7th International Conference on Computer Supported Cooperative Work in Design', IEEE, pp. 19–24.

Baker, T. & Nelson, R. E. (2005), 'Creating something from nothing: Resource construction through entrepreneurial bricolage', *Administrative science quarterly* **50**(3), 329–366.

Baskerville, R. L. (1999), 'Investigating information systems with action research', *Communications of the AIS* **2**(3es).

Baskerville, R. & Wood-Harper, T. (2002), A critical perspective on action research as a method for information systems research, *in* M. D. Myers & D. Avison, eds, 'Qualitative Research in Information Systems', SAGE Publications, Ltd, pp. 128–146.

Benson, T. (2012), *Principles of Health Interoperability HL7 and SNOMED*, Springer.

Bjørn, P. & Balka, E. (2009), Supporting the Design of Health Information Systems: Action Research as Knowledge Translation, *in* 'System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on', pp. 1–10.

Blichfeldt, B. S. & Andersen, J. R. (2006), 'Creating a wider audience for action research: Learning from case-study research', *Journal of Research Practice* **2**(1), Article–D2.

Body (2014), http://wiki.hl7.de/index.php/cdaab2:Body. [Online], accessed 1.07.2014.

Bolaños, R. A. A. & Muñoz, A. U. (2009), Implementacíon del estándard hl7 para la notifiación de eventos de intereés en salud publica, Master's thesis, Universidad del Cauca.

Boluda, L. G. (2013), Diseño e implementación de un sistema informatizado de control de "enfermedades transmitidas por vectores", Master's thesis, Universidad Rey Juan Carlos.

Boone, K. (2011), *The CDA TM book*, Springer.

Bortis, G. (2008), Experiences with mirth, *in* 'Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference on', pp. 649–652.

Braa, J., Hanseth, O., Heywood, A., Mohammed, W. & Shaw, V. (2007), 'Developing health information systems in developing countries: the flexible standards strategy', *MIS Quarterly* **31**(Special), 1–22.

Braa, J., Monteiro, E. & Sahay, S. (2004), 'Networks of action: sustainable health information systems across developing countries', *Mis Quarterly* **28**(3), 337–362.

Braa, J. & Sahay, S. (2012), *Integrated health information architecture: power for the users : design development and use*, Matrix Publishing.

Buena práctica: Sistema de Información de Salud con DHIS2 en Cauca (2013), http://www.paho.org/ict4health/index.php?option=com_bpform&view=show&fid=12&lang=en&Itemid=. [Online], accessed 28.7.2014.

Carlile, P. R. (2004), 'Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries', *Organization Science* **15**(5), 555–568.

Castrillón, H. Y., González, C. & López, D. M. (2012), 'Modelo Arquitectónico para Interoperabilidad entre Instituciones Prestadoras de Salud en Colombia', *Revista Ingeniería Biomédica* **6**(12).

Cecez-Kecmanovic, D., Janson, M. & Brown, A. (2002), 'The rationality framework for a critical study of information systems', *Journal of Information Technology* **17**(4), 215–227.

Chappell, D. (2004), *Enterprise Service Bus: Theory and Practice*, O'Reilly. EPUB.

Checkland, P. & Holwell, S. (1998), 'Action research: its nature and validity', *Systemic Practice and Action Research* **11**(1), 9–21.

Chilundo, B. & Aanestad, M. (2005), 'Negotiating multiple rationalities in the process of integrating the information systems of disease specific health programmes', *The Electronic Journal on Information Systems in Developing Countries* **20**(2), 1–28.

Ciborra, C. (2002), *The labyrinths of information*, Oxford University Press.

Ciborra, C. U., Braa, K., Cordella, A., Dahlbom, B., Failla, A. & Hanseth, O. (2000), *From Control to Drift: The Dynamics of Corporate Information Infrastructures*, Oxford University Press.

Cobo, H. Y. C. & Garcia, M. D. (2012), Proceso de Implementación de un Sistema de Gestión de Información en Sauld para el Departamento del Cauca. Unpublished.

Cuadro de temas Registro Único de Afiliados - RUAF (2013), http://www.minsalud.gov.co/salud/Paginas/RegistroÚnicodeAfiliados-RUAF.aspx. [Online], accessed 25.4.2013.

Davison, R., Martinsons, M. G. & Kock, N. (2004), 'Principles of canonical action research', *Information systems journal* **14**(1), 65–86.

DHIS2 (2014), http://dhis2.org/. [Online] accessed 28.7.2014.

Dickens, L. & Watkins, K. (1999), 'Action research: rethinking lewin', *Management Learning* **30**(2), 127–140.

Dolin, R. H., Alschuler, L., Beebe, C., Biron, P. V., Boyer, S. L., Essin, D., Kimber, E., Lincoln, T. & Mattison, J. E. (2001), 'The hl7 clinical document architecture', *Journal of the American Medical Informatics Association* **8**(6), 552–569.

Dolin, R. H., Alschuler, L., Boyer, S., Beebe, C., Behlen, F. M., Biron, P. V. & Shvo, A. S. (2006), 'Hl7 clinical document architecture, release 2', *Journal of the American Medical Informatics Association* **13**(1), 30–39.

Dossot, D. & D'emic, J. (2010), *Mule in Action*, Manning Publications Co.

Echavarría, M. Z. P. & Granda, E. J. I. (2010), 'Indicadores básicos departamento del cauca republica de colombia', PAHO.

Eclipse Instance Editor (2009), http://hl7book.net/index.php?title=Eclipse_Instance_Editor. [Online], accessed 4.06.2014.

Erl, T. (2004), *Service-Oriented Architecture. A Field Guide to Integrating XML and Web Services*, Prentice Hall.

Fichas de Notificación (n.d.), http://www.ins.gov.co/lineas-de-accion/Subdireccion-Vigilancia/sivigila/Paginas/fichas-de-notificacion.aspx. [Online], accessed 1.06.2014.

Francisco, B. & Yepes, J. (2012), 'Colombia : A Healthcare System in Crisis', *Medical Solutions* (November), 1–6.

Free Tools (n.d.), https://www.lantanagroup.com/resources/free-tools/. [Online], accessed 2.06.2014.

Fundación HL7 Colombia (2014), http://www.hl7.org.co/nuevo/index.php?option=com_content&view=article&id=47&Itemid=54. [Online]; accessed 15.02.2014.

Fuse (n.d.), http://fuse.fusesource.org/tools.html. [Online], accessed 14.05.2014.

Gilpin, M. (1999), 'Bank mergers in france highlight city planning: Lessons learned for enterprise architecture', *Giga Information Group, Ideabyte* .

Goel, A. (2006), 'Enterprise integration eai vs. soa vs. esb', *Infosys Technologies White Paper* .

Grieve, G. (2014), 'Cda use in the pcehr: Lessons learned', *Pulse It* .

Grupo Sistemas De Información en Salud Pública SIVIGILA (2014), *Manual del usario software SIVIGILA*, 3 edn, Instituto Nacional de Salud. Annex 2: Estructure de los registros para notificación de datos de eventos de interés en salud pública.

Habermas, J. (1984), 'The theory of communicative action - reasons and rationality of society', *Boston Beacon Press* .

Hanseth, O. (2001), 'Gateways—just as important as standards: How the internet won the "religious war" over standards in scandinavia', *Knowledge, Technology & Policy* **14**(3), 71–89.

Hanseth, O., Bygstad, B., Ellingsen, G., Johannessen, L. K. & Larsen, E. (2012), 'Ict standardization strategies and service innovation in health care', *Thirty third International Conference on Information Systems* .

Hanseth, O. & Monteiro, E. (1998), Understanding information infrastructure. Manuscript.

Health Metrics Network (2008), 'Framework and standards for country health information systems.'. World Health Organization.

HISP (2011), http://www.mn.uio.no/ifi/english/research/networks/hisp/. Online, accessed 29.7.2014.

HL7 FAQs (2014), http://www.hl7.org/about/FAQs/index.cfm?ref=nav. Online, accessed 15.02.2014.

HL7 Standards (2014), http://www.hl7.org/implement/standards/product_section.cfm?section=1. Online, accessed 15.02.2014.

Hohpe, G. & Woolf, B. (2003), *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Hohpe, G. & Woolf, B. (n.d.), 'Integration Patterns overview', http://www.eaipatterns.com/eaipatterns.html. [Online], accessed 1.06.2014.

Holley, K. E. & Arsanjani, A. (2010), *100 SOA Questions: Asked and Answered*, Pearson Education.

Josuttis, N. M. (2007), *SOA in Practice: The Art of Distributed System Design*, O'Reilly.

json-simple (n.d.), http://code.google.com/p/json-simple/. [Online], accessed 15.06.2014.

Kock, N. (2007), The three threats of organizational action research, *in* N. Kock, ed., 'Information Systems Action Research', Vol. 13 of *Integrated Series in Information Systems*, Springer US, pp. 97–129.

Kvale, S. & Brinkmann, S. (2009), *Interviews learning the craft of qualitative research interviewing*, SAGE.

Lewin, K. (1946), 'Action research and minority problems', *Journal of social issues* **2**(4), 34–46.

Lippeveld, T. & Sauerborn, R. (2000), Introduction, *in* T. Lippeveld, R. Sauerborn & C. Bodart, eds, 'Design and implementation of health information systems', World Health Organization, Geneva, pp. 1–14.

Liu, L. & Özsu, M. T., eds (2009), *Encyclopedia of Database Systems*, Springer US.

Lopez, D. M. (2009), Interoperable Architectures for Advanced Health Information Systems, PhD thesis, University of Regensburg.

Madden, R. (2010), *Being Etnographic*, Sage Publications.

Manouvrier, B. & Ménard, L. (2008), *Application Integration: EAI, B2B, BPM and SOA*, STE Ltd and John Wiley Sons.

Mason, R. (2011), 'ESB or not to ESB revisited part 1', http://blogs.mulesoft.org/esb-or-not-to-esb-revisited-%E2%80%93-part/. [Online], accessed 28.05.2014.

Masters, J. (1995), 'The history of action research', *Action Research Electronic Reader* . The University of Sydney. http://www.behs.cchs.usyd.edu.au/arow/Reader/rmasters.htm. [Online], accessed 21.12.2013.

McKay, J. & Marshall, P. (2007), Driven by two masters, serving both, *in* N. Kock, ed., 'Information Systems Action Research', Vol. 13 of *Integrated Series in Information Systems*, Springer US, pp. 131–158.

Meet Mule (n.d.), http://www.mulesoft.org/documentation/display/34X/Meet+Mule. [Online], accessed 29.05.2014.

Minisetrio de Salud y Protección Social (2011), 'Indicadores básicos 2011, situación de salud en colombia'.

MIRTH (n.d.), http://www.mirthcorp.com/. [Online], accessed 29.7.2014.

Mule Concepts (n.d.), http://www.mulesoft.org/documentation/display/current/Mule+Concepts. [Online], accessed 5.06.2014.

Mule Message Structure (n.d.), http://www.mulesoft.org/documentation/display/34X/Mule+Message+Structure. [Online], accessed 5.06.2014.

Myers, M. D. (1997), 'Qualitative research in information systems', *MIS Quarterly* **21**(2), 241–242.

NIKT fagforum for arkitektur and Acondo (2008), Tjenesteorientert arkitektur i spesialisthelsetjenesten, Technical report, Nasjonal IKT.

Orlikowski, W. J. & Baroudi, J. J. (1991), 'Studying Information Technology in Organizations: Research Approaches and Assumptions', *Information Systems Research* **2**(1), 1–28.

Pan American Health Organization (2012), 'Health in the americas'.

Poblacion Colombia Ahora (n.d.), http://www.dane.gov.co/reloj/reloj_animado.php. [Online], accessed 28.07.2014.

Rademakers, T. & Dirksen, J. (2009), *Open Source ESBs in Action*, Manning Publications.

Rosen, M., Lublinsky, B., Smith, K. T. & Balcer, M. J. (2008), *Applied SOA: service-oriented architecture and design strategies*, Wiley Publishing, Inc., Indianapolis, Indiana. EPUB.

Schulte, R. (1997), 'Architecture and planning for modern application styles', *Strategic Analysis Report* .

Shreshta, L. B. & Bodart, C. (2000), Data Transmissionm, Data Processing and Data Quality , *in* T. Lippeveld, R. Sauerborn & C. Bodart, eds, 'Design and Implementation of Health Information Systems', World Health Organization, pp. 128–145.

Sistema de Informacioón de Prestaciones de Sauld - RIPS (2013), http://www.minsalud.gov.co/salud/Paginas/rips.aspx. [Online], accessed 23.5.2013.

Sistema Integral de Información de la Protección Social - SISPRO (2013), http://www.minsalud.gov.co/consultaInformacion/Paginas/Home.aspx. Online; accessed 25.5.2013.

Smith, B. & Ceusters, W. (2006), 'Hl7 rim: an incoherent standard', *Studies in Health Technology and Informatics* **124**, 133–138.

Snyder, B., Rob, D. & Bosanac, D. (2009), *ActiveMQ in Action*, Manning Publications.

Spronk, R. J. & Grieve, G. (2008), 'Common issues found in implementations of the hl7 clinical document architecture (cda)', *9th International HL7 Interoperability Conference* .

Star, S. L. & Ruhleder, K. (1996), 'Steps toward an ecology of infrastructure: Design and access for large information spaces', *Information systems research* **7**(1), 111–134.

Susman, G. & Evered, R. (1978), 'An assessment of the scientific merits of action research', *Administrative science quarterly* **23**(4), 582–603.

Suárez, H. (2012), 'Preguntas Frecuentes de SISMED Sobre temas generales', http://web.sispro.gov.co. Online, accessed 27.7.2014.

Sykes, C. & Treleaven, L. (2009), Critical action research and organizational ethnography, *in* 'Organizational Ethnography Studying the Complexity of Everyday Life', SAGE Publications, Ltd, pp. 215–230.

Sæbø, J. I. (2013), Global Scaling of Health Information Infrastructures: Circulating Translations, PhD thesis, University of Oslo.

Sæbø, J. I., Kossi, E. K., Titlestad, O. H., Tohouri, R. R. & Braa, J. (2011), 'Comparing strategies to integrate health information systems following a data warehouse approach in four countries', *Information Technology for Development* **17**(1), 42–60.

Talend (n.d.), http://talend.com/. [Online], accessed 10.05.2014.

The World Factbook (2013), https://www.cia.gov/library/publications/the-world-factbook/geos/co.html. [Online], accessed 09.11.2013.

Uribe Velez, A. (2006), 'Decreto 3518: Por el cual se crea y reglamenta el sistema de vigilancia en salud pública y se dictan otras disposiciones.'. República de Colombia. Ministerio de la Protección Social.

Verjans, S. (2005), 'Bricolage as a way of life–improvisation and irony in information systems', *European Journal of Information Systems* **14**(5), 504–506.

Weber, M. (1977), *Economía y Sociedad*, Vol. 1, Fondo de la Cultura Economica.

Webster, J. (1995), 'Networks of collaboration or conflict? electronic data interchange and power in the supply chain', *The Journal of Strategic Information Systems* **4**(1), 31–42.

What is an ESB (n.d.), http://www.mulesoft.org/what-esb. [Online], accessed 17.06.2014.

Williams, T. (1997), 'Interorganisational information systems: issues affecting interorganisational cooperation', *The Journal of Strategic Information Systems* **6**(3), 231–250.

World Health Organization (1978), Declaration of alma-ata, *in* 'International conference on primary health care, Alma-Ata, USSR', Vol. 6, p. 12.