The University of Cape Town

Doctoral Thesis

Modelling evolving clinical practice guidelines: A case of Malawi

Author:
Yamiko Joseph Msosa

Supervisors:
Dr. C. Maria Keet
Dr. Melissa Densmore

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Research Centre in ICT4D
The Department of Computer Science

June 2018

Declaration of Authorship

I, Yamiko Joseph Msosa, declare that this thesis titled, ‘Modelling evolving clinical practice guidelines: A case of Malawi’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Signed by candidate

Date: 18th February 2018

Prior Publications

Some early versions of the content in this thesis appeared in the following publications:

- **Msosa, Yamiko Joseph**, Melissa Densmore, and C. Maria Keet. ”Characterisation of Clinical Practice Guideline Changes.” *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC)*. 2016.
- **Msosa, Yamiko Joseph**, Melissa Densmore, and C. Maria Keet. ”Towards an architectural design of a guideline-driven EMR system: a contextual inquiry of Malawi.” *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development.* ACM, 2015.

"An aspiration achieved without effort doesn't build wisdom."

Kentaro Toyama

Abstract

Electronic medical record (EMR) systems are increasingly being adopted in low- and middle-income countries. This provides an opportunity to support task-shifted health workers with guideline-based clinical decision support to improve the quality of healthcare delivery. However, the formalization of clinical practice guidelines (CPGs) into computer-interpretable guidelines (CIGs) for clinical decision support in such a setting is a very challenging task due to the evolving nature of CPGs and limited healthcare budgets. This study proposed that a CIG modelling language that considers CPG change requirements in their representation models could enable semi-automated support of CPG change operations thereby reducing the burden of maintaining CIGs.

Characteristics of CPG changes were investigated to elucidate CPG change requirements using CPG documents from Malawi where EMR systems are routinely used. Thereafter, a model-driven engineering approach was taken to design a CIG modelling framework that has a novel domain-specific modelling language called *FCIG* for the modelling of evolving CIGs. The CIG modelling framework was implemented using the Xtext framework. The national antiretroviral therapy EMR system for Malawi was extended into a prototype with *FCIG* support for experimentation. Further studies were conducted with CIG modellers. The evaluations were conducted to answer the following research questions: *i*) What are the CPG change requirements for modelling an evolving CIG? *ii*) Can a model-driven engineering approach adequately support the modelling of an evolving CIG? *iii*) What is the effect of modelling an evolving CIG using *FCIG* in comparison with the Health Level Seven (HL7) standard for modelling CIGs? Data was collected using questionnaires, logs and observations. The results indicated that fine-grained components of a CPG are affected by CPG changes and that those components are not included explicitly in current executable CIG language models. The results also showed that by including explicit semantics for elements that are affected by CPG changes in a language model, smart-editing features for supporting CPG change operations can be enabled in a language-aware code editor. The results further showed that both experienced and CIG modellers perceived *FCIG* as highly usable. Furthermore, the results suggested that *FCIG* performs significantly better at CIG modelling tasks as compared to the HL7 standard, Arden Syntax.

This study provides empirical evidence that a model-driven engineering approach to clinical guideline formalization supports the authoring and maintenance of evolving CIGs to provide up-to-date clinical decision support in low- and middle-income countries.

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr C. Maria Keet and Dr Melissa Densmore, for teaching me a lot and for being patient with me during this journey. I would like to extend special thanks to Melissa for always going out of her way to give me the much-needed emotional support and guidance in all those tricky and rough times. I would also like to thank Associate Professor Audrey Mbogho and the late Professor Gary Marsden for believing in me and accepting me as their doctoral student, in addition to entertaining my research ideas during the early stages of my PhD journey. I am also very thankful to Dr Brian DeRenzi for reading and commenting on some of my work, and providing me with the emotional support and guidance during the many turbulent times. I would also like to thank Associate Professor Hussein Suleman for always sparing me some time to provide guidance and advice.

I would like to thank the Hasso Plattner Institut, for the financial support they gave me during my doctoral studies.

I am also grateful to the Malawi Ministry of Health, for giving me access to a number of health facilities, health workers and technical publications. I would also like to thank the Baobab Health Trust for giving me access to their technical personnel. Such access was very invaluable to my research.

My time at UCT was very enjoyable, largely due to the many friends that I made from the Research Centre in ICT4D. James, Colins, Richard, Josiah, Joan, Bhavana, Fiona, Blessings, Lighton, Tsabi, Pacome, Mgala and Ntwa for the many long nights and days in the lab, and the many fruitful discussions, debates and presentations.

I would also like to thank Professor Lucy Maliwichi (Aunt Lucy) and Professor Wallace Chigona for proofreading some significant parts of my dissertation.

Lastly but not least, I would also like to thank my family. Especially my loving wife, Geraldine, for being very supportive and understanding throughout my PhD journey, for putting up with my countless nights that were spent at the lab, my never-ending mood swings, and my random rants about my research work. My mum and dad, for being ever present to listen to my random and often turning stories about my aspirations. My siblings Cecilia, Angela, Alan, Trinitas, Lewis and Vanessa for being such an awesome source of inspiration and strength.

Contents

List of Figures

List of Tables

Abbreviations

ANSI	**A**merican **N**ational **S**tandards **I**nstitute
AST	**A**bstract **S**yntax **T**ree
ADC	**A**rea **D**evelopment **C**ommittee
BPMN	**B**usiness **P**rocess **M**odelling **N**otation
CDS	**C**linical **D**ecision **S**upport
CDSS	**C**linical **D**ecision **S**upport **S**ystem
CHAM	**C**hristian **H**ealth **A**ssociation of **M**alawi
CIG	**C**omputer **I**nterpretable **G**uideline
CPG	**C**linical **P**ractice **G**uideline
EBM	**E**vidence-**B**ased **M**edicine
EHR	**E**lectronic **H**ealth **R**ecord
ELOC	**E**ffective **L**ines **O**f **C**ode
EMF	**E**clipse **M**odelling **F**ramework
EMR	**E**lectronic **M**edical **R**ecord
GB	**G**iga**B**yte
GHz	**G**iga**H**ert**z**
GDP	**G**ross **D**omestic **P**roduct
GVH	**G**roup **V**illage **H**eadman
HDI	**H**uman **D**evelopment **I**ndex
HL7	**H**ealth **L**evel (**7**)Seven
HIV	**H**uman **I**mmunodeficiency **V**irus
IBM	**I**nternational **B**usiness **M**achines
ISO	**I**nternational **S**tandards **O**rganization
LOC	**L**ines **O**f **C**ode
LMIC	**L**ow- and **M**iddle-**I**ncome **C**ountry

LTS	**L**ong-**T**erm **S**upport
MDE	**M**odel **D**riven **E**ngineering
MDSD	**M**odel **D**riven **S**oftware **D**evelopment
MGDS	**M**alawi **G**rowth and **D**evelopment **S**trategy
MOH	**M**inistry **O**f **H**ealth
MP	**M**ember of **P**arliament
MPS	**M**eta **P**rogramming **S**ystem
POC	**P**oint **O**f **C**are
PSSUQ	**P**ost-**S**tudy **S**ystem **U**sability **Q**uestionnaire
QUIS	**Q**uestionnaire for **U**ser **I**nteraction **S**atisfaction
SAGE	**S**tandards-based **A**ctive **G**uideline **E**nvironment
SUMI	**S**oftware **U**sability **M**easurement **I**nstrument
SUQ	**S**tandardised **U**sability **Q**uestionnaire
SUS	**S**ystem **U**sability **S**cale
TA	**T**raditional **A**uthority
UML	**U**nified **M**odelling **L**anguage
UMUX	**U**sability **M**etric for **U**ser e**X**perience
VDC	**V**illage **D**evelopment **C**ommittee
WHO	**W**orld **H**ealth **O**rganization

This thesis is dedicated to my late grandmother, Agogo Kamiya Kachenjela-Maliwichi, for always believing in me.

Chapter 1

Introduction

1.1 Background

Malawi, one of the least developed countries in sub-Saharan Africa, faces a double disease burden of communicable and non-communicable diseases [3]. In addition, the country is facing a crisis in human health resources due to a shortage of health workers [4]. Malawi has one of the lowest doctor to patient ratios estimated at about two for every 100,000 [5].

Electronic medical records (EMRs) are the future of documenting and delivering healthcare as such systems can decrease medical errors and improve quality of care [6]. The potential of EMR systems to transform medical care practice has been recognised over the past decades, including the enhancement of healthcare delivery and facilitation of decision-making processes [7]. Improving the quality of care through an EMR system depends on effective clinical decision support as a recommendation can be provided at the time of decision-making as part of the clinician's workflow, which reduces practice variation, and is based on best-practice guidelines [6].

Malawi, like most other low- and middle-income countries, has adopted task shifting to cope with the human resource crisis for health. Task-shifting refers to a process of delegation of tasks to health workers with lower qualifications [8]. The task-shifting process requires the development of standardised protocols, including simplified clinical practice guidelines (CPGs), simplified recording and reporting systems and simplified monitoring and evaluation [4]. CPG representations that allow practitioners with limited training to effectively manage the most prevalent manifestations of disease are essential for national treatment programs like those in Malawi [9]. Several attempts have been made as part of the general computerisation within healthcare to integrate CPGs in computer-supported clinical settings in form of computer-interpretable guidelines (CIGs) [10]. CPG formalization as CIGs make it possible to develop guideline-based EMR systems that have a better chance of impacting clinician behaviour than narrative CPGs [11]. It can be argued that CIGs for clinical decision support are a necessary future of medical decision making [12].

Clinical knowledge is often limited or partial, and it is not unusual for certain guidelines to be revised or proved wrong. In such cases, clinical decision support systems designers need to update their systems whenever new guidelines are introduced or old ones revised [13]. Integrating CIGs with patient-specific data and clinical applications; and executing CIGs in real-time to provide decision support and care planning still remain as challenges in healthcare [14]. Evolving requirements from the medical field combined with the properties of information systems demand advanced features in computer interpretable guidelines (CIGs) to tackle problems emanating from CPG updates [15].

Malawi like most other low-resource regions, deploys paper-based CPGs for use by task-shifted healthcare workers. But some of the CPGs have been integrated into national EMR systems for guideline-based clinical decision support. In such situations, maintenance of the EMR systems to support new and revised CPGs is a challenge. [16]

1.2 Problem statement

The aim of this work was to devise an appropriate framework for modelling and maintaining CIGs in computer-supported clinical information systems for low- and middle-income countries. In this work, I further sought to evaluate the effect of modelling and maintaining CIGs using a newly devised CIG modelling framework that is based on model-driven engineering techniques.

The scope of this study was on modelling evolving CIGs in clinical information systems that support primary healthcare delivery in low- and middle-income countries. I conducted the study within primary healthcare in Malawi. I chose Malawi because it is a low- and middle-income country that routinely uses EMR systems in public health facilities. Furthermore, Malawi is facing a double crisis of high disease burden and severe shortage of qualified health workers. This study focused on two CPGs, namely *Integrated Guidelines for the Management of HIV* and *Integrated Management of Child Illnesses*, because they are both adapted from the World Health Organisation (WHO) guidelines and are mandated for use by frontline health workers in Malawi. Frontline health workers are the first point of contact for medical care in many low- and middle-income countries including Malawi [17, 18].

1.3 Research questions

In order to address the research problem, I posed three research questions.

1.3.1 What are the CPG change requirements for modelling an evolving CIG?

In order to address the first research question, I compared successive versions of CPGs from Malawi. I analysed the changes that were identified in the comparison which resulted in their classification into ten categories of CPG changes.

1.3.2 Can a model-driven engineering approach adequately support the modelling of an evolving CIG?

I started by conceptualising a four-layer architecture for modelling evolving CIGs that is discussed in Chapter 5. This four-layer architecture has at its core, a domain-specific language (DSL) named *FCIG*, that has explicit semantics for specifying CPG structural elements that are affected by CPG changes. In order to address this second research question, I encoded a representative sample of CPGs from Malawi using *FCIG* and further evaluated *FCIG*'s language constructs for their evolving CIG representation adequacy and usability.

1.3.3 What is the effect of modelling an evolving CIG using *FCIG* in comparison with the HL7 standard for modelling CIGs?

In order to address the third research question, I conducted a study that measured the effect of modelling evolving CIGs using *FCIG*. This was done by getting novice modellers to model evolving CIGs using *FCIG* at one time and also using the HL7 standard Arden Syntax at another time. Data from the two runs of the experiment was analysed in order to measure the time on task, task success rate, error rate, number of errors encountered, efficiency, lines of code (LOC), effective lines of code (ELOC) and system usability scale (SUS) scores.

1.4 Research design and approach

In this research, I used a mixed-methods design in four studies. A mixed-methods research design allows the collection and analysing of data using a mixture of quantitative and qualitative methods in a single study or a series of studies in order to understand a particular phenomenon [19–21].

In order to address the first research question, I categorised CPG change requirements for a typical low- and middle-income country. I started by collecting CPG documents for HIV management from Malawi and elucidated CPG change requirements for a low- and middle-income country through an inductive learning process. Through this process, I identified ten categories of CPG changes that included related specifications of the CPG structural elements that are affected by the changes.

In order to address the second research question, I used the CPG change requirements for low- and middle-income country to systematically derive *FCIG*, an evolving CIG modelling language that operates within a four-layer guideline modelling architecture. I further evaluated *FCIG*'s language constructs for their representation adequacy and usability.

In order to address the third research question, I conducted a set of experiments with two experimental conditions that allowed *FCIG* to be compared with the Health Level Seven (HL7) standard for modelling CPGs, Arden Syntax.

1.5 Contributions

By addressing the research questions, this research resulted in the following contributions:

1. A systematic characterisation of CPG changes.
2. The concept of an evolving computer-interpretable guideline.
3. A four-layer computer-interpretable guideline modelling architecture.
4. A compact and usable CIG modelling language, *FCIG*.
5. Empirical evidence with regards to the effect of using the novel CIG modelling framework on CIG modellers.

I anticipate that the above listed contributions will generate interest amongst guideline-based clinical decision support systems designers and researchers, particularly those that work in low- and middle-income countries. These contributions will enhance the understanding of how sustainable guideline-based clinical decision support systems can be conceptualised and operationalised in low- and middle-income countries.

Furthermore, this research demonstrated a systematic and methodical process of designing practical and usable CIG modelling framework that is particularly relevant in a low- and middle-income countries.

1.6 Thesis outline

The rest of this thesis is structured into seven chapters as follows:

Chapter 2: Background

In this chapter, I explore and review literature relevant to this research. I further discuss the context of this work in relation to current literature. I start by discussing clinical practice guideline formalization and model-driven engineering. Thereafter, I give a background of Malawi, which is the research context. And lastly, I discuss findings from a contextual inquiry that I carried out to better understand the research context.

Chapter 3: Research design

This chapter discusses the overall approach and design to this work. I start by discussing the aims of this research, followed by their related research questions that guided this work. I further discuss the legal and ethical considerations that were taken during this research study.

Chapter 4: Characterising clinical practice guideline changes

In this chapter, I use an inductive learning approach to characterise the changes that occur when a CPG evolves over time. The characterisation in this chapter serves as a foundation for grounding CPG change requirements for low-resource settings.

Chapter 5: Modelling evolving computer-interpretable guidelines

This chapter discusses the framework for modelling evolving CIGs. The framework aims to address the CPG modelling challenges that were discussed in *Chapter 4*. I start by discussing the theoretical foundations of a four-layer CIG modelling architecture. The theoretical foundations are core to the evolving CIG modelling framework. Thereafter, I evaluate the modelling language constructs of *FCIG*, the CIG modelling language that is central to the four-layer CIG modelling architecture, for their CPG representation adequacy.

Chapter 6: FCIG grammar evaluation of perceived usability

This chapter discusses the evaluation of the language constructs of *FCIG* for their usability. I start by discussing how the evaluation was conducted with both novice and experienced CIG modellers. Thereafter I discuss the results from the evaluation and their implications.

Chapter 7: Experimental evaluation of FCIG

This chapter discusses the experiments that were carried to assess the effectiveness of *FCIG*. I start by discussing how the experiments were carried out. Thereafter, I discuss the results from the experiments in relation to the evaluation criteria that was derived to address the research questions and their related hypotheses.

Chapter 8: Conclusion

In this chapter, I start by restating the proposition from this work in relation to its research questions. I follow up with a synthesis of how the empirical results addressed the research questions. Thereafter, I discuss the contributions from this work and their implications. Finally, I discuss the limitations of this research and opportunities for future work.

Chapter 2

Background

This chapter discusses related work and sets the context for this study. Section 2.1 and Section 2.2 discuss related work, followed by a discussion that characterises Malawi in Section 2.3. The chapter concludes, in Section 2.4, with a discussion of results from a contextual inquiry that was carried out in Malawi, so as to get a deeper insight of the context for this research work.

2.1 Clinical practice guidelines

Computer-interpretable guidelines have been used to support health workers in providing care to patients in computer-supported settings. The aim of this research was to solve the challenges of maintaining clinical knowledge for automated clinical decision support using model-driven engineering techniques in low- and middle-income countries. Clinical knowledge from the best evidence is typically available for clinical practice in the form of clinical practice guidelines (CPGs) [22–25]. Therefore, this section starts with a discussion on clinical practice guidelines and their formalization for clinical decision support. CPG formalization languages can be regarded as domain-specific languages for use in the clinical domain. Hence, this section concludes with a discussion on domain-specific languages in the context of model-driven engineering.

2.1.1 Evidence-based medicine

Clinical practice guidelines are typically translated from the best evidence [23–25]. Hence, this section starts by discussing evidence-based medicine and its relation to clinical practice guidelines. Thereafter, the formalization of clinical practice guidelines into computer-interpretable guidelines is discussed as this research work focused on automated clinical decision support.

Evidence-based medicine (EBM) has been defined by its proponents as the conscientious, explicit and judicious use of the current best evidence in making decisions about clinical care of individual patients in order to provide optimal clinical care [26–30]. Practitioners of EBM go through a process of life-long and self-directed learning in which they convert information needs about diagnosis, prognosis, therapy and other health care issues into answerable questions when providing care to their patients [27]. Health workers practising EBM identify and apply the most effective clinical interventions in order to maximise the quantity and quality of life for individual patients [28].

EBM is particularly relevant to low- and middle-income countries where the global burden of disease and illness is primarily situated [31–37]. If EBM is good for richer countries, then the case is stronger in low- and middle-income countries [35–37] as it is essential to know which interventions work and which ones do not [32]. Low- to middle-income countries need appropriate technology such that their healthcare delivery systems have practitioners and patients that have access to the most appropriate evidence they deserve [32, 33].

Across most domains in medicine, findings consistently show that practice lags behind knowledge that is usually available in the form of clinical practice guidelines [22]. Evidence exists that many CPGs, even those that are broadly accepted, are not followed [22, 38, 39]. A core part of practising evidence-based medicine is considering CPGs when they do exist [22]. Although EBM has had many benefits, it has also had some negative outcomes as the evidence-based quality mark has been misappropriated by vested interests [40–42]. Although this research did not focus on validation of the source of evidence that is used in CPG development, it contributes towards filling of this gap by promoting the adoption of national CPGs in low- and middle-income countries that typically use validated best evidence. National CPGs in low- and middle-income countries are usually adapted from evidence-based recommendations from World Health Organization (WHO) CPGs [43, 44]. Another drawback of EBM is that the volume of evidence in form of clinical practice guidelines has become unmanageable in clinical practice [42, 45]. These reasons have necessitated the need to refocus on providing usable evidence that can optimise care given to patients [42]. This asserts the need to evermore experiment with usable guideline modelling frameworks that can support automated clinical decision support and the maintenance of the computer-interpretable guidelines deployed in such an environment.

2.1.2 Clinical practice guidelines in low- and middle-income countries

Clinical practice guidelines are systematically developed statements that include recommendations to assist practitioner and patient decisions in order to optimise patient care for specific clinical circumstances [46, 47]. The term *Clinical Practice Guideline* can serve as an umbrella label for practice standards, protocols, parameters, algorithms and various other types of statements about appropriate clinical care [48].

CPGs are developed to support the introduction of new knowledge and assist health practitioners in managing patients in order to improve the quality of care in clinical practice. CPGs describe the evidence-based procedures to be followed during diagnosis, treatment and decision making for a specific disease based on years of accumulated medical experience [49]. The fundamental value of CPGs is to ensure that tasks are carried out uniformly in addition to serving as a guide or reminder in situations in which it is likely that steps will be forgotten, are difficult to follow, or where errors can be expensive [50]. CPGs aim to improve the quality of care, reduce unjustified variations and reduce healthcare costs [11]. Furthermore, CPGs contribute an important modality that can reduce the delivery of inappropriate care and support the introduction of new knowledge into clinical practice [51].

While CPGs assist health practitioners in managing their patients, there are several factors that make it difficult for the health practitioners to follow and adhere to often complex CPGs. CPGs can become a source of stress and increased complexity for clinicians [52]. CPGs can also be difficult to use in practice as they can include tens or even hundreds of pages of complex information, vocabulary and diagrams dealing with multiple aspects of care and treatment [49]. Further to that, several factors limiting or restricting complete adherence to clinical guidelines include lack of awareness with the guidelines existence, lack of agreement, lack of physician self-efficacy, lack of outcome expectancy or the inherent difficulty to change habits in daily behaviour [53].

CPGs that are intended for developing countries differ from those that are developed for use in richer countries. CPGs for many conditions in low- and middle-income countries are designed for use by health workers with limited training that may also have limited access to diagnostic testing tools [54]. The WHO, by virtue of being a technical resource for health systems globally, develops CPGs that can be adapted and adopted by individual countries in the developing world for use in their national treatment programs such as Malaria and HIV [31, 54, 55]. WHO guidelines are usually based on strategies that are safe, simple, and inexpensive in order to support scaling up of national treatment programs [56, 57]. These CPGs are often revised as new evidence emerge on how to manage the various conditions targeted in the CPGs [57–59]. This public health approach leads to a high diagnostic sensitivity at the expense of specificity [60].

2.1.3 Computer-based clinical practice guidelines

Computer-based CPGs have been developed to facilitate the integration of clinical guidelines into clinical environments that are equipped with digital technology [49]. Lyng [10] argues that three approaches have been promoted for computerising CPGs as part of the general computerisation efforts within healthcare: *i*) where computer-based CPGs are promoted as modelling languages independent of their execution environment, *ii*) where computer-based CPGs are promoted as an execution environment, *iii*) where computer-based CPGs are integrated into an Electronic Health Record (EHR) system.

Computer-interpretable guidelines (CIGs) are formalized models of CPGs that make it possible develop guideline-based clinical decision support systems (CDSSs) [11, 12]. Furthermore, modelling CPGs in a computer interpretable form is a pre-requisite for various computer applications to support CPG application [61]. A CIG modelling formalism should be supported by a formal language that has a complete set of vocabulary, syntax and semantics [12]. The current and emerging standards for expressing CPGs have a formal syntax but no formal semantics as there are no mathematical models that define what specifications in these languages mean [11]. Furthermore, these CIG modelling languages and frameworks that aim at improving CPG application in computer-supported settings are insufficient as they rely on informal processes and notations [62]. The work presented in this thesis aimed at filling this gap by proposing a usable CIG modelling framework that has a formal notation and clear semantics for expressing a CIG.

A CPG's formalization into computer-based guidelines is necessary for automated medical decision support. CPG formalization as CIGs make it possible to develop CIG-based CDSSs which have a better chance of impacting clinician behaviour than narrative guidelines [11]. In addition, the medical community has started to recognise that CIGs can further increase CPG advantages by providing relevant benefits such as automatic linkage to patient data and decision support to care providers and patients [63]. Guideline-based CDSSs are deemed necessary for the future of medical decision making in general [12].

There are a number of challenges that make it difficult to formalize a CPG into a CIG. Data and knowledge modelling of CPGs is a relatively neglected area, yet it has enormous impact on the format and expressiveness of decision criteria, and the ease with which guidelines can be formalised and integrated into decision-support services [64]. In addition, the way data and medical concepts are represented and linked to Electronic

Medical Record (EMR) systems have resulted in incompatibilities among CPG encodings [64–67]. This asserts the need to follow a knowledge modelling approach that can leverage an existing medical vocabulary when formalizing CPGs. Furthermore, existing tools for encoding and maintaining CIG models are limited in their ease-of-use and the support they offer to the CIG modeller [68]. The work presented in this thesis aimed at filling this gap by proposing a usable guideline modelling framework that leverages existing medical dictionaries when modelling CIGs.

Computerisation of CPGs for guideline-based CDSS at the point-of-care (POC) has the potential to improve clinicians' adherence to CPGs in routine clinical practice. The automation and computerisation of the daily management of both clinical guidelines and patient data can lead to improvement of physicians' adherence to clinical guidelines and makes this a basic step towards widespread use in medical practice [53]. Moreover, implementing guidelines in active computer-based CDSSs promises to improve guideline adherence because these systems are able to monitor the actions and observations of care providers and to provide guideline-based advice at the POC [12]. This research investigated the potential of addressing this gap by providing a comprehensive framework for integrating a CIG into an Electronic Medical Record system to encourage CPG application at the point-of-care.

The formalization of CPGs into CIGs is a complex task that can be categorised into several approaches. Each approach for specifying computer-based guidelines has its own motivations and features as some focus more on guideline standardisation and interoperability while others focus more on guideline development and decision support [11]. This formalization of CPGs can be classified into two main categories. The first category is the document-based approach that uses the CPG document as a medium of representation based on markup text using markup languages like XML [11, 69]. Examples of the document-based approach are Hypertext Guideline Markup Language (HGML) [70] and Guideline Elements Model (GEM) [71]. The document-based approach was not appropriate for this study as this study focused on executable CIGs that can be integrated with clinical information systems to provide patient-specific recommendations. The second category, which is the focus of this research, is whereby knowledge is extracted from CPGs and represented as executable constructs in a specific format [69]. Examples of the executable CIG constructs are Arden Syntax for Medical Logic Modules [72–74], GuideLine Interchange Format (GLIF) [75], Asbru [76] or PROforma [77]. The executable CPG formalization approaches can be further categorised into two kinds of methods: *i*) case-based reasoning or chaining of individual rules; *ii*) task-network models, where a CIG execution engine executes a CPG that is represented as a network of component tasks that unfold over time [11].

For the rest of this thesis, the term CIG shall mean an executable CIG. There are a number of formalisms for expressing CIGs [11, 12, 78, 79]. Though most of the CIG formalisms have advanced the state of knowledge on how CPGs can be represented in computer interpretable format, they are experimental and lack proven validity in practice [79]. Practical and technical limitations force researchers developing guideline modelling formalisms and execution software to confine the use of their technology to their home institutions [80]. The work in this thesis aimed at addressing this gap by proposing an evolving CIG conceptual model that can serve as a basis of interoperable CIG modelling frameworks. The various existing CIG modelling formalisms are discussed in the sections that follow.

2.1.3.1 Arden Syntax

Arden Syntax for Medical Logic Modules (MLMs), established in 1989 and subsequently developed as a Health Level Seven (HL7) certified standard [73, 79], is a framework for providing decision support in EMR systems [78]. Arden Syntax represents procedural clinical knowledge in MLMs. Each MLM contains sufficient knowledge to make a single decision that invokes a specific action [78]. Arden Syntax has been used for clinical decision support by generating clinical alerts, diagnostic interpretations, management messages and screening for research studies [81–83]. Though Arden Syntax has been used for clinical decision support in clinical information systems, it sometimes requires a clinical information exporting mechanism which introduces a substantial system load with delays in event detection and data provision [82]. In addition, Arden Syntax requires a specialised Arden Syntax engine to execute MLMs within a clinical information system [83]. Though Arden Syntax supports a more human readable writing style [83], it poses considerable challenges when it is required to integrate it with existing heterogeneous clinical information systems [84]. This research considered these challenges by proposing a specialised CIG domain-specific language (DSL) within a layered and lightweight model-driven engineering framework that can facilitate CIG integration with heterogeneous clinical information systems.

2.1.3.2 EON

The EON model, introduced in 1996, came out of research carried out of Stanford University [85]. The model is a component-based architecture for building protocol-based decision support systems for therapy-planning. Using this model, system builders define a domain ontology for concepts in the application area, configure a problem-solving method that implements some control strategy, and map the concepts defined in the domain ontology to the data requirements of a problem solving method. The researchers developed generic problem solvers for determining patients' eligibility for protocols and for establishing appropriate protocol-directed therapy [86]. EON uses a task-based approach to define decision-support services using alternative solutions that can be selected from a toolkit [87]. However, the EON model is inappropriate to the public health approach to CPGs that is typically adopted by low- and middle-income countries as the EON model requires knowledge of a particular institutional clinical workflow.

2.1.3.3 Prodigy

The Prodigy model was developed at the University of Newcastle upon Tyne. It provides support for chronic disease management by facilitating knowledge engineering through a simple and understandable model that is sufficiently expressive [88]. The model divides a clinical guideline in two sections. The first, describes the management of a chronic disease over time and demonstrates available therapeutic choices for each recognisable state of the disease being managed. The second, describes actions that may be taken each time the patient is seen by a clinician. A high-level view of a guideline is represented as a network of scenarios, action steps and subguidelines. Clinical interventions are split into actions, which are instantaneous as far as primary care is concerned, and activities, interventions that persist until they are modified or stopped. Criteria are implemented as Boolean expressions to express preference for or against a choice of action steps, scenarios or prescription items [89]. However, this model just like other process-centric CIG modelling languages is inappropriate for a public health approach to CPGs which was the focus of this research.

2.1.3.4 Standards-based Active Guideline Environment (SAGE)

Standards-based Active Guideline Environment (SAGE), developed and introduced in 2002 by a consortium consisting of research groups at GE Healthcare, University of Nebraska Medical Center, Apelon Inc., Stanford Medical Informatics and Mayo Clinic, is a technology platform for integrating guideline-based decision support into enterprise clinical information systems [90]. The project builds on previous guideline modelling work that includes Asbru, GEM, GLIF, EON, PROforma, GUIDE and PRODIGY. The researchers in the SAGE project argue that for effectiveness of clinical decision support to be maximised, a clinical decision support system must be invoked at an opportune moment in the clinical care process and must facilitate clinical workflow non-intrusively [80]. A central concept of SAGE, characterised by a triggering event, patient characteristics, organisational setting, roles specifying who responds to an event, and needed resources, is context that coordinates the activation of guideline-based decision support [11]. Guideline recommendations are modelled as either activity graphs representing guideline-directed processes or decision maps that represent recommendations at a point in time [80]. SAGE still remains an experimental platform whereby SAGE-encoded guidelines have not been used in practice [79]. To address this gap, the work in this thesis aims at creating an acceptable CIG DSL to novice CIG modellers with the aim of increasing its likelihood of being adopted.

2.1.3.5 PROforma

PROforma, an executable process modelling language [77], was developed at the Advanced Computational Laboratory of Cancer Research in the UK [79]. It is essentially a first-order logic formalism extended to support decision making and plan execution [79]. In PROforma, a guideline is modelled as a set of tasks and data items that are organised hierarchically into plans [77]. Tasks are divided into four classes [77, 91]. Actions represent a request for an external actor to do something, Enquiries represent points at which information is requested from an external actor, Decisions are points at which choices need to be made from a set of candidates [91], and Plans are collections of tasks grouped together for a particular reason [77]. PROforma as process modelling language does not suit the context of the work in this thesis. The work in this thesis focuses on public health approach CPG formalization that necessitates no knowledge of institution-specific clinical workflows such as those required when modelling CIGs with PROforma.

2.1.3.6 GUIDE

GUIDE, introduced in 1998, is a component-based and multi-level architecture developed by the Laboratory for Medical Informatics in Italy [79]. This methodology allows CPG implementation by means of a careflow management system [92]. The model allows representation and execution of CPGs with both workflow management systems and EMR systems [79], extending a CPG implementation from a simple reminder to an organiser of healthcare processes [93]. In contrast, the work in this thesis aimed at a public health approach to CPG application which requires little sensitivity to institution-specific clinical workflows.

2.1.3.7 Guideline-Interchange Format (GLIF)

Guideline-Interchange Format (GLIF), developed and introduced in 1998 by members of Intermed Collaboratory at the Universities of Columbia, Havard, McGill and Stanford, is a structured computer-interpretable language for modelling and executing CPGs [75, 79, 94]. The current version, GLIF3, allows a formal definition of decision criteria, action specifications and patient data [75]. The model enables CPG encoding at three levels: a conceptual flowchart, a logically verifiable and computable specification, and an implementable specification that can be incorporated into a particular institutional information system [75, 95]. GELLO, an HL7 standard object-oriented expression language, is included in GLIF as a formal expression language for specifying decision criteria and patient states [11, 96]. While GLIF is a more comprehensive guideline representation ontology than most CIG models, it remains an experimental language with inadequate support for automated execution and it lacks proven capacity of integration with institutional information systems [79]. The work in this thesis aimed to fill this gap by proposing a comprehensive CIG modelling architecture with integration and execution capabilities.

2.1.3.8 GASTON framework

The GASTON framework, introduced in 1997, was developed at the University of Maastricht in the Netherlands [97]. The framework uses a guideline representation formalism that combines a number of approaches based on the concepts of primitives, problem solving methods (PSMs) and ontologies [11, 98]. The primitives construct a CPGs control structure explicitly [98] based on Guideline Interchange Format (GLIF) version 2.0 [11]. The PSMs are used to model CPGs that perform stereotypical tasks [98]. The ontologies are used to model the medical domain [11]. The GASTON framework is unsuitable for a public health approach to CPG application as it requires knowledge of institution-specific clinical workflows to model CIGs.

2.1.3.9 Asbru

The Asgaard project led by Vienna University of Technology and Stanford Medical Informatics introduced Asbru in 1997. Asbru is a task-specific, intention-based and time-oriented language for representing skeletal plans [76]. The researchers in the project view clinical practice guidelines as a set of schematic plans for the management of patients who have a particular condition. The researchers further argue that, when guidelines are applied, a care provider interprets skeletal plans that have been designed by the guidelines author and that providing support to the guideline application process implies an interactive process [99]. Asbru is also unsuitable for the context of the study in this thesis as it requires knowledge of institution-specific clinical workflows.

2.1.3.10 Other CIG modelling approaches

Colombet et al. [100] and Seroussi et al. [101] propose the usage of decision trees and probabilistic models to formalize CPGs into CIGs, where the optimal action strategy is evaluated though a decision tree [11]. Microsoft Excel and Visio were used as editing tools to author and visualise the decision trees respectively [100]. In contrast, the work presented in this thesis differs in its focus as it aimed at proposing and evaluating CIG modelling tools that can overcome the burden of authoring and maintaining oft-changing CIGs.

Pérez and Porres [102] propose a different approach to CPG formalization by providing a framework that aims at enabling authoring and verification of CPGs using model checking and model-driven development techniques to verify CPGs against semantic errors and inconsistencies in their definition. The Unified Modelling Language (UML) and Statecharts were used to represent the dynamics of each CPG [102]. In contrast, this study follows a rule-based approach to CIG modelling with a specialised domain-specific language for modelling CIGs that follow a public health approach.

Kamsu-Foguem et al. [103] propose the use of a graph-based approach to CPG formalization so that sound logical reasoning can be supported in a visual manner to enhance CIG understandability by those that have no formal background. A conceptual graph is constructed from a collection of a formal and detailed collection of nodes, relations and questions [103]. In contrast, the focus of the work in this thesis targeted the formalization of CPGs that are applied in a public health manner thereby requiring no knowledge of institution-specific clinical workflows.

Semantic web technologies have been proposed as a method for formalizing CPGs [14, 104, 105]. A knowledge management approach can be taken to ontologically model practice guidelines in terms of their content, structure and function. This can be done by creating an ontology that entails a semantic abstraction of practice guidelines in terms of practice-related knowledge structural elements and relationships between elements [14]. Hurley and Abidi [14] define a clinical pathway ontology that describes the structure and function of clinical pathways. Abidi et al. [106] further extend the previous work into a clinical decision-support framework for handling comorbidities by combining and aligning ontology modelled CPGs using web ontology language (OWL). Abidi et al. [106] use semantic web rule language (SWRL) rules to define all conditions related to the merging process, increasing the effort required to maintain resulting systems and reducing the possibility of sharing knowledge [15].

Huang et al. [107] also used semantic web technologies to represent evidence-based CPGs in order to address the interoperability barrier to adopting CIGs. They represent CPGs using Resource Description Framework (RDF) and OWL [107]. Building on previous work, Zamborlini et al. [108] specify a formal model for detecting interactions among recommendations within CIGs. Zamborlini et al. [108] represent guidelines using RDF graph structures in a semantic web framework for representing and reasoning about guideline recommendations and beliefs building on the work of Hoekstra et al. [109], Huang et al. [107] and Mons et al. [110]. But semantic web technologies such as RDF, SWRL and OWL can be cumbersome to the CIG modeller when authoring and maintaining CIGs.

Business Process Modelling Notation (BPMN) has also been used to formalize CPGs. Kaiser and Marcos [111] propose an approach that encodes CPG knowledge using BPMN and semi-automatically transforms the encoded CPG into an executable CIG using languages such as Asbru and PROforma. Rodriguez Loya et al. [112], Braun et al. [113] and Aziz et al. [114] used BPMN to formalize and execute CPGs in their work. BPMN requires knowledge of institution-specific clinical workflow processes which does not suit the public health approach to CPGs that is typically adopted in low- and middle-income countries.

2.1.4 Support for computer-interpretable guideline evolution

Peleg [11] argues that capacity for managing the complexities involved in supporting CIGs that evolve over time should be provided in CPG formalization systems. There have been a number of efforts that have tried to address the complexities involved in supporting evolving CIGs [115–117].

Miller et al. [115] analysed the nature of the domain knowledge that is required to perform automated validation after updates are applied to CIGs for child-hood immunisation in the United States of America. The patterns identified by Miller et al. [115] for possible automation cannot be generalised as they were specific to child-hood immunisation CIGs in the United States of America.

Attempts have also been made to incorporate version management capabilities in CIG systems to facilitate CIG maintenance for evolving CPGs. There are two approaches to CIG versioning that have been proposed in literature. The first, is the model-centric approach, where a versioning tool is used to support the creation of new or the modification of existing CPGs by tracking change operations of basic structural components of a CIG model [116, 117]. The second, is the document-centric approach, where CPG changes are tracked in the original guideline document and systematically applied to its corresponding formal or semi-formal CIG model [61].

Peleg and Kantor [118] propose logical models for version management of CIGs represented in GLIF. The intention of models that were proposed by Peleg and Kantor [118] for version management is to ensure that reasoning functions are related to correct guideline recommendations in different versions of a CPG, that may be required by different users at a single point in time [116].

Grandi et al. [117] propose a method of version management that enables multi-version representation of a CPG to allow access to multiple temporal perspectives of a CIG version that can be tailored to a specific use case. In this method, the DeGeL framework is used to specify CPGs in different formalisms such as ASBRU, GEM and GLIF [117, 119]. Using the DeGeL framework, multiple representations of a CPG are provided at different representation levels from semi-formal to formal [117].

Kaiser and Miksch [61] propose a different method of version management by updating previously formalized CPGs in a series of steps that apply the changes from textual form into Asbru using a document-centric approach. Though the version management approaches proposed by Peleg et al. [116], Grandi et al. [117] and Kaiser and Miksch [61] address how to access different versions of a CIG at a single point in time from a single repository, their conceptual models do not precisely define what changes and how a particular type of change should be handled within a CIG model when a new version of a CPG is introduced.

2.1.5 Electronic Medical Record systems in low- and middle-income countries

Guideline-based clinical decision support systems are a promising tool that can improve healthcare delivery in task-shifted settings [120–122]. Clinical guideline formalization for clinical decision support is normally discussed in the context of health informatics. Health informatics can be defined as an interdisciplinary field that studies and supports the effective use of biomedical data, information and knowledge for scientific inquiry and to support healthcare and healthcare delivery for individuals, families, groups and communities [123–125]. Health informatics draws from several scientific disciplines including computer science, information science and health science disciplines due to its interdisciplinary nature [123, 124]. The terms ‘biomedical informatics’ and ‘medical informatics’ have been used as alternative names due to the evolving nature of the discipline over time and the perception others have had on the term 'health informatics' as one that does not reflect the field's broad range of applicability by excluding fields such as basic human biology [124, 125]. In this thesis, I consider the terms 'health informatics', 'biomedical informatics' and 'medical informatics' to be synonymous in line with the previously given definition of the term 'health informatics'.

An Electronic Medical Record system is the backbone of health informatics. An EMR system is a set of computational tools that are used for managing a repository of electronically maintained information about an individual's health status and healthcare [124]. The term 'Electronic Health Record (EHR) system' has been used as an alternative term to 'EMR system' though others have perceived an EMR system to be an inter-organisational system [126, 127]. In this thesis, I consider the terms 'EHR system' and 'EMR system' to be synonymous. An EMR system is central to patient care services as it contains active information management tools that can provide clinical reminders and alerts, and linkages with clinical knowledge sources for clinical decision support [123, 124, 128]. These active information management tools can contain CPG formalization tools which are a focus of this research.

Various EMR systems have been deployed in many low- and middle-income countries with the aim of improving healthcare delivery [129, 130]. The EMR systems have been adopted in these countries to support national programs for primary care and other specialist treatment categories such as HIV, Tuberculosis, Maternal and Child Health, Diabetes and Immunisation [131–136].

EMR systems are increasingly being equipped with functionality that can increase their uptake. A number of researchers and practitioners have worked on patient identifier management in EMR systems to support the continuity of care in low- and middle-income countries [122, 135, 137, 138]. The usage of barcodes and local language searching algorithms has also been proposed to support patient searching functionality in EMR systems for low- and middle-income countries [122, 138].

Free and open source EMR systems are recommended in low- and middle-income countries due to limited budgets for healthcare in addition to supporting software development and maintenance capacity [139–142]. Though free and open source EMR systems are recommended and gaining ground in low- and middle-income countries, there still remains a need to improve the skills that are needed to implement and use these systems so that the realisation of their potential benefits can be maximised [143].

There are two categories of medical records that can be accessed from an EMR system for clinical decision support. The first category pertains to those medical records that are collected on paper during a patient's visit at a health centre and subsequently entered into an EMR. Such a method is called retrospective data entry normally undertaken by non-clinical personnel such as data entry clerks [138, 144, 145]. Though it has been argued that retrospective entry limits clinical decision support capabilities in EMR systems [122, 135], others have argued that delivery of care can still be improved through clinical decision support provided through clinical summaries [146–148]. The second category is where medical data is entered into an EMR system by a care provider during a patient's visit at a health centre. Such a method is called point-of-care data entry [122, 135, 149]. An EMR system that uses the point-of-care data entry method can be called a provider-entry or point-of-care EMR system [122, 150]. Point-of-care clinical decision support in EMR systems can also improve the quality of care as patient-specific advice can be given during a clinical visit [122, 151]. The availability of EMR systems in low- and middle-income countries provide a unique opportunity that can be exploited to improve healthcare delivery through efficient and effective guideline-based clinical decision support.

2.2 Model-driven engineering

Model-driven engineering (MDE) refers to software development approaches in which abstract models of software are created and systematically transformed into concrete implementations. In MDE, the primary artefacts of development are prescriptive models [152, 153]. A prescriptive model is a model that is rigorous, formal, complete and consistent such that it can be used to automatically construct a target system [153]. The main engineering principle behind it, is that the inherent complexity of software development can be mastered by building, analysing and manipulating system models [154]. This approach allows the reduction of the gap between problem and software implementation domains through the use of technologies that support systematic transformation of problem-level abstractions to software implementations [152]. MDE allows the creation of formal, tool-processable representations of specific aspects of software systems using a domain-specific language (DSL) that allows the representations to be transformed into executable code [153]. CIG modelling languages can be regarded as DSLs that provide basic abstractions that are required to model CIGs [155, 156]. The work in this thesis experiments with a CIG modelling DSL that enables guideline-based clinical decision support through an MDE approach in a low- and middle-income country. To focus the discussion on MDE, the discussion is segmented in two parts: *i*) Domain-specific languages; *ii*) Model-driven domain-specific language development.

2.2.1 Domain-specific languages

A DSL is a computer programming language of limited expressiveness that is focused on a particular class of problems, called a domain [153, 157]. DSLs sacrifice some of the flexibility to express any program in a general-purpose language (GPL) for productivity, ease of use, and conciseness of relevant programs, with corresponding gains in productivity and reduced maintenance costs, in a particular domain [153, 158]. The process of building a DSL can help deepen an understanding of a DSL and its domain [153]. DSLs address issues of low-level syntax and lacking of expressiveness by enabling developers or domain experts to program at a high-level abstraction [159]. Therefore, by building a DSL that is specifically tailored towards CPG formalization in low- and middle-income countries, one can gain a deeper understanding of the implications of formalizing CPGs using a CIG DSL in low- and middle-income countries.

Many computer programming languages are domain-specific and have been called many names over the years [158] such as application-oriented [160], special purpose [161], forth-generation [162], little [163], task-specific [164] and specialised [165] languages.

Although DSL development is hard, requiring both domain knowledge and language development expertise [158], designing and implementing a small DSL from scratch is often relatively easy, enabling innovative designers to rapidly develop high-impact solutions [166]. A DSL specifically tailored towards CPG formalization in low- and middle-income countries can be developed with the potential of revolutionising clinical decision support in regions that are burdened by disease.

2.2.2 Model-based domain-specific language development

There are several case studies and experience reports in literature that present DSL development experiences using a model-driven engineering approach [167]. These reports can be categorised into three groups. The first, report on experiences and lessons learnt from development of individual projects (e.g. [168–170]). The second reports on DSL development in the context of model-driven software development (MDSD) (e.g. [171–174]). The third, report on particular techniques for specific aspects related to the design and implementation of DSLs (e.g. [158, 170, 175–180]).

Voelter et al. [153] argues that the development of a DSL primarily requires the composition of an abstract syntax, a concrete syntax, constraints and type systems, and execution semantics. But the systematic development of a DSL can be achieved by carrying out activities from the following categories: *i*) Definition of scope and purpose of the DSL, *ii*) Definition of domain abstractions and core language model, *iii*) Definition of the behaviour of language elements, *iv*) Definition of DSL concrete syntax, *v*) Integration of DSL artifacts with an information system platform, *vi*) Evaluation and refinement of DSL [1, 153, 167]. The activities within a DSL development process should be tailored to the nature of the intended DSL. The activities within such a process are typically carried out iteratively as requirements become clearer during development [1]. Figure 2.1 illustrates a typical model-driven DSL development process.

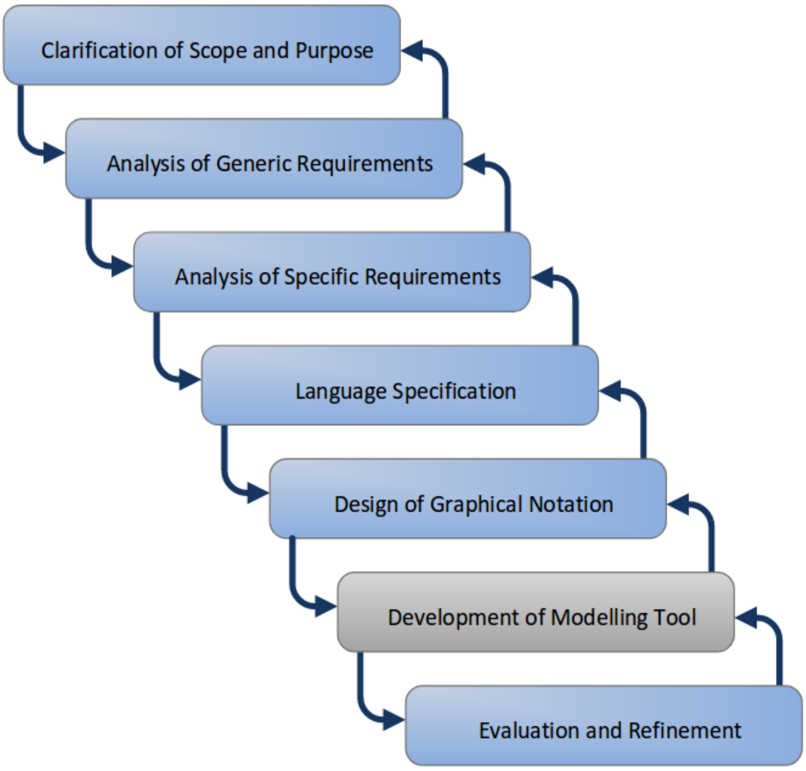

Clarification of Scope and Purpose
Analysis of Generic Requirements
Analysis of Specific Requirements
Language Specification
Design of Graphical Notation
Development of Modelling Tool
Evaluation and Refinement

Figure 2.1: An illustration of a model-driven DSL development process [1]

In practice, DSL development is exploratory whereby different DSLs are typically developed in different application contexts. Hence, the order in which activities are performed and the exact steps with their associated guidelines that must be taken can vary widely [1, 167]. In such cases, DSL designers should reflect upon the various guidelines and adapt them if required [1]. Hence the general approach to DSL development consists of two major tasks. The first, involves tailoring the DSL engineering process to the context of a particular project. The second, involves applying the tailored DSL engineering process to develop the DSL [167].

There are three main models of the types of DSL engineering processes [167]. The first, language model driven approach, is whereby the language model definition drives the DSL development. The second, mockup language model, is a DSL development process that starts with concrete syntax design in order to raise the participation of domain experts. The third, extracting a DSL from an existing system approach, entails a process whereby domain abstractions are extracted directly from an existing system. Kosar et al. [179] argue that many DSL development communities advocate their DSL implementation approaches and work on improving their techniques and supporting tools.

A software language is any language that is created to describe and create software such as programming, modelling and domain-specific languages [181]. DSLs can be regarded as specification languages or programming languages because DSLs are usually declarative [182]. Since DSLs are software languages, existing software language development methods can be used on DSLs.

A formal software language can be developed by specifying its formal language model and implementing a concrete syntax that conforms to the semantics of the language model. The section that follows discusses software language models of DSLs.

2.2.3 Formal language model specification

A core language model of a particular DSL is created by identifying domain abstractions and defining language model semantics. A language model contains language concepts that are intended as abstractions over types of a particular conceptual model [1]. The core language model semantics, also known as static semantics, express invariants on language model concepts and relations between the concepts. The semantics of a language concept should be invariant throughout a domain of interest and in time [167]. The static semantics consist of rules that dictate whether or not an expression of the language is well formed or not [183].

Constraints and type systems can be used to validate static semantics of a particular program. It can be argued that not all programs that conform to the structure of a language are valid. Constraints aim to solve this problem by enabling specification of restrictions that cannot be expressed purely by structure [153]. These constraints are boolean expressions that must be true for every program expressed in a particular language. Constraints can be implemented with any language that is able to query a model and report errors to the user [153]. On the other hand, a type system is a collection of type rules for a typed software language such as a DSL [184]. Type systems are a subset of constraints. The type systems associate types with program elements and then check whether these types conform to pre-defined typing rules [153].

Another form of semantics that can manifest when using a DSL are execution semantics also known as dynamic semantics. Execution semantics denote the observable behaviour of a program as it is executed [153]. These semantics define the effects that result from using a language concept. The behaviour of language elements can be specified though a DSL behaviour specification which can take a form that ranges from high-level control flow models to precise textual specifications [167]. Execution semantics add another layer of meaning to the language concepts that are defined in the abstract syntax of a DSL [183]. The observable behaviour of a DSL can be established with a sufficient number of tests, or with model checking and proof [153].

The abstract syntax of a DSL is a data structure that holds the core information in a program, but without any of the notational details contained in the concrete syntax [153]. An abstract syntax specification defines the basic notions of model elements in a language and their relations in a mathematically precise way, with structural constraints, multiplicities and implicit relationships [185]. On the other hand, the concrete syntax of a DSL is what the user interacts with to create programs [153]. A concrete syntax specification targets the actual visual appearance of a language by assigning a visual symbol to model elements in a language [185].

Some notable work has been undertaken to propose formal theories for software language specification. Software language specifications can be constructed from theories over types. Reynolds [186] and Harper [187] propose a set of theories that can be used for formal software language specification based on predicate logic. Turner [188] also propose a conceptually similar set of theories that are based on typed predicate logic. Noting that DSLs are software languages, existing software language theories can be used in the formal specification of language models for DSLs.

Once a formal language model for a particular DSL has been established, a concrete syntax that can be used by a domain expert should be developed. The section that follows discusses the specification of a DSL's concrete syntax.

2.2.4 Concrete syntax specification

A concrete syntax is a DSL's interface that represents the abstractions defined in its abstract syntax. This interface should be convenient for human users as well as being easy to process by software components [167]. Most language specification experts focus on semantics and abstract syntax, paying little care and consideration to the design and specification of concrete syntax [1]. There are two major categories of DSL concrete syntax: *i*) Textual DSLs that use linear textual notations based on ASCII, Unicode or other extended set of symbols; *ii*) Graphical DSLs that use graphical shapes such as box-and-line diagrams [153, 157, 167]. DSLs are easily adopted by their user community when existing or familiar notations are adopted. Hence, a good choice of concrete syntax is important for DSLs to be accepted by the DSL users [153].

A DSL's concrete syntax can be implemented using a language workbench. A language workbench is a software engineering tool that helps its users to efficiently create a language and enable integration with the tools to create models in these languages [153, 157, 189]. Language workbenches reduce the effort required to develop DSLs and their IDEs [153]. Fowler [157] argues that a language workbench can be treated as a parser when used to define concrete syntax with its model-aware tooling against an existing semantic model. This prevents a semantic model from being locked in a specific language workbench [157]. There are several language workbenches that have been proposed to support the creation of DSLs [153, 157, 190, 191].

Erdweg et al. [191] compared ten language workbenches according to their language engineering features that were classified into six categories: *i*) Notation – these are features that determine how programs or models are presented to users; *ii*) Semantics – these are features that support specification of language semantics; *iii*) Editor support – these are feature that support syntactic and semantic editing of programs or models; *iv*) Validation – these are features that support language-specific validation that are semantic in nature; *v*) Testing – these are features that support the debugging of the language definition and the construction of the debuggers; *vi*) Composability – these are features that support extension of existing languages and unification of multiple languages into a single language. They concluded that language workbenches provide adequate abstractions for implementing DSLs though the extent at which individual language workbenches support language engineering features vary widely [191]. Stoffel [192] compares three separate language workbenches according to their features. In another study, Vasudevan and Tratt [190] conducted a comparative analysis of four language workbenches for internal textual DSLs. Ribeiro et al. [193] also conducted a comparative analysis of three language workbenches for UML profile-based languages using qualitative criteria in separate study. These studies came to a conclusion that language workbench features vary widely such that selection of a particular language workbench should be based on the nature of the project at hand [190, 192, 193]. Though language workbenches vary in terms of their features and function, they present us with a unique opportunity that can be exploited by supporting the efficient implementation of a CIG DSL with its related tooling for authoring and maintaining CIGs.

Metaedit+, initially proposed by Smolander et al. [194] out of the MetaPHOR research laboratory, is a commercially available language workbench that can be used for implementing DSLs [195, 196]. Though Metaedit+ has been widely used in the industry for over two decades [197], it uses a proprietary structure definition that makes the models and languages constructed with not portable to other language workbenches and editors [198]. A further drawback of Metaedit+ is that it is a commercial platform that requires a pricy licence to operate.

Meta Programming System (MPS) is an open source language workbench, being developed by Jetbrains, that supports many advanced features for creating graphical DSLs [153, 189]. MPS structure definition is proprietary though the language workbench is open source. Voelter et al. [153] argues that MPS' structure definition does not implement any acceptable industry standard though it is conceptually close to [199] Ecore, a model structure definition standard of the Eclipse Modelling Framework (EMF). Since MPS language workbench is limited to the creation of graphical DSLs [200], it may not suit textual DSL use cases such as support for rule-based CIG modelling. Another drawback of the MPS language workbench is that it currently only supports tooling for the IntelliJ IDEA IDE as it uses a proprietary structure definition [153, 201].

Spoofax is an open source Eclipse-based language workbench, developed out of the University of Delft, that can be used for creating textual DSLs and tooling for use with the Eclipse IDE [153, 202]. Spoofax uses a structure definition called A Term that is based on a generic tree structure whose textual notation can be used for exchanging data between tools [153]. Spoofax lacks a mechanism that can easily reuse implementations by referencing existing DSL definitions [192].

Intentional Domain Workbench (IDW) is a commercially available language workbench that can be used for creating graphical DSLs that is conceptually similar to MPS [153, 203]. Not many details about IDW are available due to its commercial nature [204].

Xtext is a widely adopted open source language workbench for creating textual DSLs [153, 205, 206]. Xtext uses EMF [199] open standards for its structure definition which allows its resulting DSL definitions and tooling to be highly interoperable with existing IDEs such as Eclipse and Intellij IDEA [153, 207]. Xtext may not be as advanced as MPS but its Eclipse Modelling Platform ecosystem avails a large number of addons that enable sophisticated DSL environments [153]. Xtext can be exploited when creating a CIG DSL due to its open source nature and the usage of open standards in its DSL structure definitions.

A DSL needs to be properly integrated into its target information system platform in order to be useful to its user community. The section that follows discusses the methods that can be used to integrate a DSL with its target information system platform.

2.2.5 DSL integration

DSL artifacts can be integrated with their target information systems platform by mapping DSL artifacts to platform features, extending platform to support new features, and defining a DSL execution environment through DSL-to-platform transformations or interpreters [153, 167]. In model-driven software development, DSL models can be transformed into other models that tools can use to generate code through a process called model transformation [153, 208]. Alternatively, DSL models can be transformed into text such as source code or configuration files through a process called code generation [153, 174]. A completely different approach can be taken whereby an interpreter traverses an AST of a DSL model to directly perform actions that correspond to the contents of the AST via a process called interpretation [153].

2.2.6 DSL evaluation

DSL users need to know the inherent strengths and domain applicability of the languages they use [209]. Since DSLs fall under the software language umbrella, applicable software language evaluation methods can be used for DSL evaluation. Evaluation criteria of software languages can be classified into four categories as follows: *i*) Application domain; *ii*) Human factors; *iii*) Software engineering; *iv*) Language design and implementation [209].

Application domain evaluation concerns, also known as the philosophy of design [210], entail such criteria that assess how well a language supports developing software for a specific type of applications [211, 212]. Incompatible abstractions of the problem domain between software language users and software language engineers are a constant challenge to language usability [213]. Software language users need to know the inherent strengths and domain applicability of the languages they use, in order to use the software languages effectively [209].

Human factors come into play noting that software engineering is human-intensive, as well as that such criteria can be used to assess the user-friendliness of a particular DSL [214]. Factors such as naturalness and readability of software languages have not been consistently endorsed by evidences in software language engineering [215]. DSLs can be regarded as user interfaces as they provide a domain-expert with an interface to a desired computation platform [216]. Like other software languages, DSLs have rarely received usability or human factors evaluation which may have led to the deployment of inadequate and unusable languages [217].

Software engineering concerns are those aspects that enable the development of good software by evaluating a software language's capacity to support such qualities as reliability, portability, re-usability and maintainability [209, 218]. A number of researchers have argued that DSLs can improve desirable attributes of resulting software [182, 219].

Language design and implementation concerns are those aspects that assess how well a language is designed to enable support of modelling tools such as editors, compilers and interpreters [220]. Modelling tools for a particular DSL are key to the DSL's acceptance in its modelling community [221, 222]

Several studies have investigated quality characteristics of DSLs that can guide DSL evaluations. Though there have been some studies on DSL assessment, there is currently no agreed standard for DSL evaluation [223]. Kolovos et al. [224] argue that characteristics that are used for evaluating general-purpose programming languages can also be applied when evaluating DSLs though their relative importance may differ. Kahlaoui et al. [225] propose an approach that identifies a set of success factors that be transformed into a project-specific set of assessment criteria for DSL evaluation. Kärnä et al. [226] propose a DSL evaluation approach that evaluates a DSL against pre-set goals through user experiments. Mohagheghi and Haugen [227] also propose a set of DSL assessment criteria based on their experiences in practice. Haugen and Mohagheghi [228] propose a multi-dimensional framework for characterising DSLs using a questionnaire across three dimensions: expressiveness, formalization and transparency. Wu et al. [229] propose a slightly different approach that can determine the level of effort required in developing and using a DSL through metrics that can measure the effort. Karsai et al. [180] propose a set of DSL design guidelines that can be considered when developing a DSL and classified the guidelines using development phases as follows: *i*) Language Purpose – guidelines for the early activities of the DSL development process; *ii*) Language Realization – guidelines which discuss DSL implementation methods; *iii*) Language Content – guidelines which focus on characteristics of language elements; *iv*) Concrete Syntax – guidelines that contribute to readability of a language; *v*) Abstract Syntax – guidelines that affect the internal representation of a language. Frank [1] proposes generic requirements of a DSL as follows: *i*) Concepts of a DSL should correspond to concepts that are familiar to prospective DSL users; *ii*) A DSL should provide domain-specific concepts so long as their semantics are invariant within the scope of the DSL's application; *iii*) A DSL's concepts should allow for modelling that is sufficient for all foreseeable applications; *iv*) A DSL should provide concepts that can clearly distinguish different abstraction levels within a model; *v*) A clear mapping should exist between a DSL's concepts to the concepts of the relevant target representation. DSL developers can evaluate a particular DSL against specific requirements that are derived from the generic requirements [1]. Kahraman and Bilgen [230] propose a qualitative framework for assessing DSLs that guides the selection of key DSL quality characteristics to guide an evaluator. Barišic et al. [216] propose an experimental evaluation model for DSLs that focuses on usability and is based on techniques for the experimental evaluation of user interfaces. These previous studies highlight domain-appropriateness, simplicity, usability, formality and expressiveness as important criteria that should be considered when creating a DSL to ensure that the resulting DSL is of high quality and that it can increase its likelihood of being accepted by its users.

Perceived usability can be regarded as an important high-level construct of usability [231, 232]. The subjective measurements of usability and the objective measurements of efficiency and effectiveness provide a comprehensive construct of usability [233, 234]. The International Standards Organization (ISO) defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [235, 236]. The next section discusses standardised perceived usability measurement tools.

2.2.7 Standardised usability questionnaires

Standardised usability questionnaires (SUQs) are designed to assess a participant's satisfaction with the perceived usability of products during or immediately after usability testing [231]. Hence, an SUQ can be used to assess the perceived usability of a software language. An SUQ usually undergoes psychometric qualification that reports measurements that can determine its reliability, validity and sensitivity [231, 237].

Nunnally [237] and Lewis [238] agree that among several other advantages, an SUQ can offer the following benefits: *i*) Objectivity – a standardised measurement can allow measurement statements of other practitioners to be independently verified by others; *ii*) Replicability – an SUQ can make it easier to replicate a particular study through the usage of standardised methods; *iii*) Quantification – notwithstanding that the application of statistical methods to multipoint scale data is a point of controversy, standardised measurements allow usability practitioners and researchers to report results in finer detail that use mathematical and statistical methods so that results can be better understood; *iv*) Economy – in spite of the substantial amount of work that is required when developing standardised measures, the measures in an SUQ becomes economical to use once they have been developed; *v*) Communication – standardised measures allow usability practitioners and researchers to efficiently and effectively communicate their usability results; *vi*) Scientific generalisation – SUQs provide a platform for standardisation that is essential for assessing the generalisability of results.

Several SUQs have been introduced over the past few decades [231, 238, 239]. These SUQs have been introduced primarily to assist in the assessment of usability following participation in scenario-based usability tests [231]. SUQs can be classified into two categories. The first category of SUQs includes those that administered immediately following the completion of a task or test-scenario within a study [231, 240, 241]. The second category of SUQs includes those that are used for administration at the end of a study [231, 241]. A posttest SUQ can be used as a post-study questionnaire in a study that evaluates a CIG modelling language. The rest of this section discusses some of the post-test SUQs that have undergone a psychometric evaluation process.

The questionnaire for user interaction satisfaction (QUIS) is a post-study questionnaire that was first published in 1988 [242]. The original version of QUIS had 90 items that used a 10-step bipolar scale that was numbered from 0 to 9, and was aligned with the negative response on the left in addition to an off-scale *'NA'* response to indicate non-availability [231]. A psychometric evaluation of the early version of the QUIS was reported for a short-form version of the questionnaire with 27 items covering usability aspects related to software, screen, terminology and system information, learning, and system capabilities [231, 242]. The evaluation showed evidence of construct validity and sensitivity although items in screen factor did not group as expected [231]. The current version of the QUIS is available in five languages (English, German, Italian, Portuguese, Spanish); in two lengths, short with 41 items and long with 122 items; and using a nine-point bipolar scale with an off-scale *'NA'* response similar to the initial version [231]. A licence for the QUIS is required before usage of the questionnaire as follows: *i*) A student licence requires 50 United States Dollars (USD50.00); *ii*) An academic or non-profit license requires 200 United States Dollars (USD200.00); *iii*) A commercial licence requires 750 United States Dollars (USD750.00) [231].

The software usability measurement inventory (SUMI) is a post-study SUQ that was introduced in the 1990s by the Human Factors Research Group at the University College Cork in Ireland [243, 244]. The SUMI is a 50-item questionnaire with a global scale based on 25 of the items five subscales for measuring efficiency, effectiveness, helpfulness, control and learnability [231, 244]. The SUMI contains a mixture of positive and negative statements for its items that are measured on a three-scale step namely: Agree, Undecided, and Disagree [244]. The SUMI is currently available in 12 languages (Dutch, English, Finnish, French, German, Greek, Italian, Norwegian, Polish, Portuguese, Swedish, and Spanish) [231]. SUMI underwent a considerable amount of psychometric evaluation which provided some evidence of construct validity and sensitivity [231, 244]. Licences are required to allow one to use SUMI as follows: *i*) One thousand Euros (EUR1000.00) for offline usage; *ii*) Five hundred Euros (EUR500.00) for online usage; *iii*) No fee for students [231].

The post-study system usability questionnaire (PSSUQ) is a post-study SUQ that was introduced in the early 1990s as an internal project at International Business Machines (IBM) Corporation [245]. The current version of the PSSUQ is a 19-item questionnaire with a single global scale and three subscales that measure system quality, information quality and interface quality [246]. The items in PSSUQ are scored using values between one and seven, with lower scores indicating a higher degree of satisfaction [245, 246]. The PSSUQ underwent a considerable amount of psychometric testing that provided some evidence of construct validity and sensitivity [245–247]. The PSSUQ does not require any licence fee for its use [231].

The system usability scale (SUS) is a post-test SUQ that was developed in the 1980s [232, 248]. The SUS is 10-item questionnaire that is scored using a five-step scale [248]. The scores from an SUS questionnaire are used to derive a single global scale [231, 248]. The scores from an SUS questionnaire can also be used to reliably compute two subscales that can measure usability and reliability [249]. Like other more prominent SUQs in both research and practice, the SUS underwent a considerable amount of psychometric evaluations [231, 250]. The 10 SUS items were selected from a pool of 50 potential items [231, 248]. The psychometric tests provided evidence of reliability, sensitivity and validity [231]. The SUS does not require a licence fee as a prerequisite for its use [231, 248].

The usability metrics for user experience (UMUX) is a recently developed four-item post-test SUQ that is based on a seven-point Likert scale designed to provide results on a single global scale that is similar to SUS [251]. The UMUX is organised around the ISO9421-11 definition of usability [252]. The UMUX has a shorter form variant UMUX-LITE which only has two items that align it to the technology acceptance model [252, 253]. The UMUX has undergone limited psychometric evaluation. The validity, reliability and sensitivity of UMUX as an SUQ has been criticised [253–255]. A recent study that was conducted to evaluate the validity, reliability and sensitivity of UMUX provided some evidence of construct validity and sensitivity of the UMUX questionnaire [252].

2.3 Context: Malawi

The sections that follow describe Malawi, which characterise the context of this research work. The geography and population of Malawi are described in Section 2.3.1. That is followed by a discussion of Malawi's health sector in Section 2.3.2.

2.3.1 Geography, population and the economy of Malawi

Malawi is a landlocked country in the sub-Saharan Africa. The country covers an area of approximately 118, 484 square kilometres, of which, 94, 276 square kilometres is land. The country was colonised by the British from 1, 891 until its independence in 1964.

Hence English is the official language in the country. The Central African country is 901 kilometres long, and 80 to 161 kilometres wide. Malawi is bordered by Mozambique on both sides in the south, Tanzania in the north east, and Zambia in the north west. [256] [257]

Malawi has an agro-based economy due to the fact that the country lies along the Rift Valley. In the valley, and along the entire length of the country, lies Lake Malawi, which has the Shire River draining water from it in the south. Hence there are fertile plains and mountain ranges that stretch along the west and south of Lake Malawi, making the region suitable for agricultural activities. [256]

Malawi is divided into three main regions. The regions are referred to as the northern, central and southern regions in the north, central and south of Malawi respectively. The regions have a total of 28 districts. The districts are administratively subdivided into traditional authorities (TAs) that are managed by chiefs. Each TA has a number of villages, which are the smallest administrative units that are managed by village headmen [256]. In some instances, a group village headman (GVH) oversees several villages. A village development committee (VDC), at GVH level, is responsible for development activities. Development activities at TA level are coordinated by the area development committee (ADC). Politically, each district is further divided into constituencies which are represented by Members of Parliament (MPs). In some cases, these constituencies can combine more than one TA [3].

Malawi has a tropical continental climate. The country's rainfall and temperatures vary depending on altitude and proximity to the lake. From May to August, Malawi experiences its winter with cool and dry weather. From September to April, Malawi experiences its summer with hot weather. And the rainy season begins around October and continues until April [256]. Rainfall intensity is considered a key determinant of pathogenic micro-organisms such that transmission of many parasitic diseases is confined to the rainy season [258–260].

Malawi has a population of approximately 17.2 million people with an intercensal population growth rate of 3.1% per year [5]. The country has a number of urban centres with the four major ones being Blantyre and Zomba in the southern region, Lilongwe in the central region and Mzuzu in the northern region. The percentage of urban population is just above 15 percent, the rest being rural population [256]. The country has a moderate life expectancy at birth of 64.8 for females and 62.9 for males [5]. Almost half of the population is under 15 years of age and the dependency ratio rose from 0.92 in 1966 to 1.04 in 2008 [3, 261]. About 7% of the population are infants aged less than 1 year, 22% are children under five years of age and about 46% are aged 18 years and above.

Malawi is predominantly a Christian country (83%), while 13% are Muslim, 2% of other religions and 2% of no religion [3].

In 1994, Malawi adopted a National Population Policy, which was designed to reduce population growth to a level compatible with Malawi's social and economic goals. The policy aims to reduce population growth, improve health and well-being, to improve education, and to increase employment opportunities [256]. In the same year, the country also adopted a strategy to eradicate poverty, the Malawi Growth and Development Strategy (MGDS) which was an initial five-year strategy that is now in its third instalment as MGDS III [256, 261]. The MGDS is the overarching development strategy for the country [261, 262].

Malawi has 13 African ethnic groups with the Chewas followed by the Nyanjas being the biggest groups. Linguistically, each ethnic group has its own language. However, the country has 16 listed living languages with English as the official and statutory national working language. The majority of Malawians speak Chichewa, which is the national language. [263]

The economy of Malawi is based primarily on agriculture, which accounts for 30 percent of the gross domestic product (GDP). The country's major exports are tobacco, tea, and sugar. They account for approximately 85 percent of Malawi's domestic exports. Malawi, one of the poorest countries in Africa, has a GDP per capita of $1113. The country is ranked at 170 out of 188 countries on the human development index (HDI). Human development is about equal life chances for all that involves expanding capabilities of people's present choices - to live healthy, productive and safe lives. The available statistics indicate that over ten million people live in multidimensional poverty. Over 61% live on less than $1.25 a day. [256] [261] [5] [257]

2.3.2 The health sector

In Malawi health care services are delivered by both the public and the private sectors. The public sector includes all facilities under the Ministry of Health (MOH) and the Ministry of Local Government and Rural Development, those of other ministries such as Education, and the Police, the Prison Service and the Army. The private sector consists of private-for-profit and private not-for-profit providers, mainly Christian Health Association of Malawi (CHAM). The public sector provides services free of charge while the private sector charges user fees for its services. [3, 257]

The MOH is a government agency that sets the agenda for health in Malawi in collaboration with other stakeholders. The agency is responsible for developing, reviewing and enforcing health and related policies for the health sector; spearheading sector reforms. It also regulates the health sector including the private sector in addition to developing and reviewing standards, norms and management protocols for service delivery and ensuring that these are communicated to lower level institutions. Further to that, the MOH is also responsible for planning and mobilizing of health resources for the health sector; advising other stakeholders on health-related issues; providing technical support for supervision; coordinating research; and monitoring and evaluation. [3]

The MOH established five Zonal Offices. The role of the Zonal Offices is to provide technical support to District Health Management Teams (DHMTs) in the planning, delivery and monitoring of health service delivery at the district level. The Zonal Offices further facilitate central hospitals' supervision of districts. [3]

Malawi has a high disease burden characterised by high prevalence of communicable diseases, maternal and child health problems, and increasing burdens of non-communicable and neglected tropical diseases. The Essential Health Package (EHP) that includes the delivery of health services in Malawi, covers diseases and conditions affecting the majority of the population and especially the poor. These services are offered free of charge to Malawians. The services in the package are: vaccine preventable diseases; acute respiratory infections (ARIs); malaria; tuberculosis; sexually transmitted infections (STIs) including HIV/AIDS; diarrhoeal diseases; Schistosomiasis; malnutrition; ear, nose, and skin infections; perinatal conditions; common injuries; disability and mental illness; and non-communicable diseases including hypertension and diabetes. [3] [257]

The health services are delivered at different levels, namely primary, secondary, and tertiary. These different levels are linked to each other through a comprehensive referral system that has been established within the health system. [3]

The primary care level in Malawi consists of community initiatives; health posts; dispensaries; maternity facilities; health centres; and community and rural hospitals. Health surveillance assistants (HSAs), community-based distributing agents, village health committees and other volunteers provide most of the services at the community level. In particular, the HSAs provide promotive and preventive health services such as immunization services. In some instances, the HSAs have been trained and provide services in community case management of diseases such as acute respiratory infections (ARIs), diarrhoea and pneumonia among children under five years of age. Services at the community level are conducted through door-to-door visits, village clinics and mobile clinics. [3]

Community health nurses and other health cadres also provide health services at the community level through outreach programs. Village health committees promote primary health care activities through community participation and they work with HSAs on preventive and promotive health services such as hygiene and sanitation. [3]

At primary level, health centres support the HSAs. Each health centre has a health centre advisory committee, and in collaboration with village health committees, helps communities to demand the quantity and quality of services that are expected from health centres. Health centres provide both curative and preventive healthcare services. And at a higher level, community hospitals also known as rural hospitals, provide both primary and secondary care with an admission capacity of 200 to 250 beds. [3]

The secondary level of health care is delivered from district hospitals in each district throughout the country. With an admission capacity of 200 to 300 beds, the district hospitals are referral facilities for both health centres and rural hospitals. The district hospitals also deliver primary care services to the local town population. In addition to the district hospitals, CHAM hospitals also provide secondary level health care services. The district or CHAM hospitals provide general services, primary healthcare services and technical supervision to lower healthcare service delivery units. District hospitals also provide in-service training for various personnel in the provision of healthcare services for community-based health programs. Health services within each district are managed by the district health management team at the district hospital, which receives direct technical support for supervision from Zonal Offices. [3]

This level consists of central hospitals that offer specialised services and in turn provide referral health services for district hospitals in their respective regions. The four central hospitals are Queen Elizabeth in Blantyre with 1250 beds, Kamuzu in Lilongwe with 1200 beds, Mzuzu in Mzimba District with 300 beds and Zomba in Zomba District with 450 beds. Tertiary care is also provided by Zomba Mental Hospital in Zomba. Queen Elizabeth and Kamuzu Central Hospitals are also teaching hospitals, with links to the College of Medicine and Kamuzu College of Nursing. Although the central hospitals also provide healthcare services which ought to be delivered by district health services, they are responsible for professional training, conducting research and providing support to districts. There are plans to establish gateway clinics at all central hospitals in order to decongest them. The gateway clinics will be run by the district health officers. [3]

CHAM, a non-profit health services provider, is the biggest partner for the MOH. It provides healthcare services and trains health workers through its health training institutions. CHAM owns 11 of the 16 health-training institutions in Malawi, most of them located in rural areas. CHAM facilities charge user fees to cover operational costs and are also mostly located in rural areas. The Government of Malawi heavily subsidizes CHAM by financing some Essential Medicines and all local staffing costs in CHAM facilities. [3]

The Malawi Traditional Medicine Policy guides the practice of traditional medicine in Malawi. The health sector works with traditional healers through the Malawi Traditional Healers Umbrella Organization (MTHUO). Although the relationship between the MOH and traditional healers has been weak, the MOH and stakeholders in the health sector have mainly engaged with traditional birth attendants (TBAs) with the intention of expanding maternal and child health (MCH) services to the community. [3]

2.3.3 Electronic medical record systems in Malawi

The MOH in Malawi is increasingly deploying EMR systems for routine clinical practice in both government and private health facilities with support from implementing partners and donors. There have been significant efforts by the MOH to deploy the EMR systems to support care delivery in primary care [136].

Malawi has developed EMR systems that support health workers through treatment, diagnosis and cohort-reporting for a number of national treatment programs such as antiretroviral therapy [122] and diabetes [264]. These disease-specific EMR systems are designed to support health workers in-line with national clinical practice guidelines [122][264].

2.4 Contextual inquiry in Malawi

EMR systems integrated with CPGs are increasingly being deployed to support healthcare delivery in Malawi. To devise a lasting solution for CPG formalization, it was essential to understand the challenges of operationalising guideline-based clinical decision support in EMR systems for Malawi. The challenges for Malawi were investigated by means of a contextual inquiry with 13 health professionals in seven health facilities, nine EMR system developers, and some of the implementation details of the national EMR system in Malawi. [16]

2.4.1 Methods

One tertiary, one referral and five primary care health facilities were selected in Malawi for observation of routine clinical practice. The observations were carried out for one month in July 2014. Clinical encounters were observed during patient visits and ward rounds for outpatients and inpatients respectively. In addition, semi-structured interviews were carried out with the health workers that were observed.

Nine members of the software development team that develop and maintain the national EMR systems for Malawi were also observed. In addition, semi-structured interviews were carried out with the members of the EMR system developers that were observed. Furthermore, code inspections of the national EMR systems were carried out through a publicly accessible open source repository [1]. The national ART and diabetes EMR systems for Malawi were accessed and inspected.

2.4.2 Findings

Eighty minutes' worth of interviews were recorded with thirteen healthcare workers in the seven health facilities we observed. Of these healthcare workers, five were medical doctors that had spent a minimum of six years in medical school; three were nurses that had spent a minimum of three years in nursing school; and five were medical assistants that had spent a minimum of a year in health sciences school. All of the healthcare workers that were interviewed indicated that they use clinical practice guidelines during routine clinical practice. Of the seven health facilities that were observed, four had a version of the national EMR system deployed in at least some part of the health facility. 50.7% of the healthcare workers we interviewed indicated that they had used computers since medical or nursing school. 30.8% of the healthcare workers indicated that they use digital devices or computers routinely to support them during clinical consultation.

Eighty two minutes' worth of interviews were also recorded with the members of the software development team. Of the nine team members I observed and interviewed, four were team leaders. The software developers that were interviewed had implemented CPG rules in an EMR before and had varying experiences developing guideline-driven EMR systems. From the interviews and the code inspections, it was indicated that there was no uniform way of encoding CPG rules into the EMR systems. Furthermore, the CPG rules that were encoded, were either hard-coded in software programs or encoded as records in the database depending on developer preference. All the EMR system developers found maintenance of CPG rules within the EMR systems challenging. The developers indicated that CPGs are revised on a regular basis which necessitates updates to the corresponding rules in the EMR systems. For instance, the Malawi clinical guidelines for managing HIV services were revised in 2008, 2011 and 2014, warranting significant efforts in maintaining the national ART EMR. One developer said that *'changes in guidelines require a deep understanding of the software implementation and it is hard to maintain guidelines that apply across multiple EMR systems or multiple versions of a particular EMR'*. The developers also indicated that they are given very tight deadlines to deliver working software. For instance, one software developer stated that *'we get*

[1] https://github.com/BaobabHealthTrust

unrealistic demands from end-users' and another developer further said that *'we are given unrealistic deadlines by product owners'*. The developers preferred that CPG rules be separated into their own layer or component on the software stack for ease of maintenance. The raw data from the contextual inquiry is available in Appendix I.

2.4.3 Discussion

As Malawi seeks to scale up application of computerised guidelines, the current approach that has been adopted requires a lot of software maintenance effort likely to make it an unsustainable practice. In collaboration with the EMR system developers, health workers, and other stakeholders, a viable direction would be one that allows generic specification of CIGs for later integration with EMR systems. Such a direction may enable generalised CPG support in EMR systems that would make it easier to support new and revised CPGs for clinical decision support. The failure of large complex systems such as EMR systems in meeting their deadlines, costs, and stakeholder expectations, are not by and large failures of technology, rather, these projects fail because they do not recognise the social and organisational complexity of the environment in which the systems are deployed [265]. Implementing a sustainable guideline-based EMR system needs a thorough understanding of the context such systems operate in.

Component-based software engineering is the process of defining, implementing, and integrating or composing loosely coupled, independent components into systems [266]. As digital healthcare ecosystems are becoming larger and more complex, component-based software engineering is becoming an important software development approach that allows EMR systems to be built from reusable components [267, 268]. One way that digital healthcare can cope with complexity and deliver better software more quickly is to reuse rather than reimplement software components. In the same fashion, a CIG can be specified once using a specialised CIG modelling environment that can transform the CIG into an EMR platform specific CIG using model-driven engineering techniques.

Separating the CPG rules and their execution engine into a separate component of the software architecture can improve the maintainability of the EMR systems. A healthcare system adapting to the effects of growing expenditures and a diminishing primary workforce needs the support of a flexible information infrastructure that facilitates innovation in wellness, healthcare and public health [269]. Clinical guideline recommendations can be encoded in a uniform way making it easier for CIG modellers to identify and update specific recommendations every time corresponding national CPGs are revised. The clinical guideline recommendations could be interpreted by a separate component, a guideline execution engine or evaluator, that could be part of the overall EMR architecture. A key characteristic of this approach to software design is that the system components should be not only interoperable but substitutable [269].

Another key characteristic of a component-based approach to software design for a flexible health information infrastructure is that the platform should be built to open standards to facilitate customisation, extension and innovation [269]. Open standards show promise as a means to achieving the true potential of EMR systems in improving healthcare [270]. Any open standards initiative to develop EMR systems would draw on the same ethos of peer review and open discovery that drives much of the research component of the health industry [271][272]. An open standard CIG modelling platform would allow the EMR system developer community to adopt, extend and integrate the platform into their respective EMR systems fairly easily.

2.5 Chapter summary

This chapter has discussed related work to clinical guideline formalization for computer-supported clinical decision support. The chapter further discussed the factors that characterise Malawi as a context of this study. The last part of the chapter discusses results from a contextual inquiry that was carried out in Malawi to gain a better understanding of the context. The related work and the contextual inquiry highlighted some gaps that motivated this research study. The gaps are summarised as follows:

i) There is need to support CPG application in task-shifted settings

ii) Most CIG formalization languages do not suit the modelling of public health CPGs

iii) Most existing CIG formalization environments are complex and still experimental, hindering the likelihood of their adoption by CIG modellers

iv) There is inadequate support for evolving CIGs for clinical decision support in low- and middle-income countries

. These gaps implied that there was a need to create a usable and effective evolving-CIG modelling framework that can facilitate the provision of guideline-based clinical decision support in low- and middle-income countries. Therefore, the next logical question was to ask what characterises CPG changes in low-resource settings. The literature review also highlighted some opportunities that could be explored. The opportunities are summarised below:

i) The CIG modelling framework could consider a model-driven engineering approach

ii) Evaluation of the CIG modelling framework could consider the CIG modelling language's usability

iii) An experimental approach to the CIG DSL evaluation could be adopted

iv) The CIG modelling framework could leverage existing open standards in clinical information systems that are widely adopted in low- and middle-income countries

v) Existing software language specification theories could be adopted during the CIG DSL design phase

vi) A post-study standardised usability questionnaire could be used when evaluating the novel CIG DSL

vii) The development of the CIG DSL proposed in this thesis could use the Xtext modelling framework as it is based on open standards

viii) Existing usability evaluation approaches can be adopted during the CIG DSL evaluation phase

ix) The CIG DSL proposed in this thesis could be evaluated against the current HL7 standard for modelling CIGs

. Taking into account the afore summarised gaps and opportunities, as presented previously in Section 1.3, the following research questions were posed:

1. What are the CPG change requirements for modelling an evolving CIG?
2. Can a model-driven engineering approach adequately support the modelling of an evolving CIG?
3. What is the effect of modelling an evolving CIG using *FCIG* in comparison with the HL7 standard for modelling CIGs?

. The research in this thesis aimed to address these research questions. The detailed methodologies for each of the four studies in this research discuss how the opportunities summarised in this section were exploited. The next chapter discusses the research approach that was undertaken in this study in order to answer the research questions.

Chapter 3

Research design

The work in this thesis was conducted over a series of four separate but related studies. In this chapter I present and describe the overall approach that I adopted to carry out the research presented in this thesis.

3.1 Research questions

The aim of this work was to devise an appropriate framework for modelling and maintaining CIGs in computer-supported clinical information systems for low- and middle-income countries.

In order to address the research problem, I formulated the following research questions:

1. **RQ1** – *"What are the CPG change requirements for modelling an evolving CIG?"*
2. **RQ2** – *"Can a model-driven engineering approach adequately support the modelling of an evolving CIG?"*
3. **RQ3** – *"What is the effect of modelling an evolving CIG using FCIG in comparison with the HL7 standard for modelling CIGs?"*

3.2 Research approach

I developed and applied this research in the context of public health facilities in Malawi. I used a mixed-methods approach that drew on a combination of methods that included formal specification, software language design, experimental evaluation, and quantitative and qualitative data analyses.

I started by characterising changes that occur in CPGs using clinical guideline documents from Malawi. After characterising the CPG changes, I conceptualised a novel evolving CIG modelling architecture that enables computational support for managing CPG changes. At the centre of the CIG modelling architecture, I designed a domain-specific language *FCIG* that conforms to an evolving CIG conceptual model that explicitly specifies the elements that are affected by CPG changes. Finally, I evaluated *FCIG* and its modelling architecture for its adequacy, usability and effectiveness when modelling evolving CIGs.

I segmented the research into four separate but related studies focusing on specific concerns towards answering the research questions presented in Section 3.1. The detailed methodologies for each of the studies are detailed in the respective chapters that discuss the individual studies.

In order to answer the first research question **RQ1**, I carried out a study that elucidated and categorised clinical practice guideline change requirements for low- and middle-income countries in the context of Malawi. The detailed methodology and results from that study are discussed in Chapter 4.

In order to answer the second research question **RQ2**, I carried out two studies that focussed on the following: *i*) The second study evaluated whether *FCIG* could be used to model an evolving CIG. The second study further evaluated whether FCIG could directly support fine-grained CPG changes. The methodology and the results from the study are detailed in Chapter 5; *ii*) The third study evaluated whether the language constructs of *FCIG* were usable. The methodology and the results from the study are detailed in Chapter 6.

In order to answer the third research question **RQ3**, I carried out a study that compared *FCIG* with the HL7 standard Arden Syntax when modelling evolving CIGs. The methodology and the results from the third study are detailed in Chapter 7.

3.3 Legal and ethical arrangements

I obtained ethical clearance for the research with approval number FSREC 018-2014 from the Science Faculty at the University of Cape Town. I further obtained ethical clearance for the research with approval number NHSRC # 1293 from the National Health Sciences Research Committee in Malawi. Furthermore, I obtained formal authorisation to visit and carry out research work at specific health facilities from the Lilongwe District Health Office. Refer to Sections A.1–A.4 of appendix A for copies of the approval and authorisation letters.

As the research involved working in the electronic health records domain, in all instances where patient data was required, I used anonymised retrospective patient records to protect the privacy of the patients and maintain confidentiality of the data.

3.4 Chapter summary

This chapter has outlined the approach that was undertaken to conduct this research. The research was structured into four separate but related studies. Each of the four studies, together with their detailed methodologies, are discussed in chapters that follow.

Chapter 4

Characterising clinical practice guideline changes

4.1 Introduction

In this chapter, I characterise CPG changes. I take a bottom-up inductive learning approach to analyse successive sets of CPGs to elucidate and characterise the changes that occur over time.

4.2 Materials and methods

I began by requesting and obtaining copies of guideline documents from the Central Monitoring and Evaluation Division (CMED) of the Malawi Ministry of Health. I requested only the guideline documents that had been used in the past or were still in use by healthcare workers in public health facilities.

After obtaining the guideline documents, I selected a set of guideline documents spanning at least three successive versions of CPGs. I examined each guideline document within the set to identify and extract candidate guideline recommendations. Thereafter, I encoded the extracted candidate guideline recommendations into Guideline Elements Model (GEM) using GEM Cutter version 3.0. I selected GEM Cutter because it formalizes CPGs into a structure that is close to the structure of published CPG text from guideline documents through a documentary approach to CPG modelling [69]. I encoded all candidate guideline recommendations whose conditions and actions were clearly specified in the originating guideline document. I did not encode other clinical facts and supporting information for CPGs.

After extracting and encoding the candidate guideline recommendations, I compared the first and the second version of CPGs from the selected set. Thereafter, I repeated the process by comparing the second and third versions of the CPGs. For each comparison cycle, I adopted an inductive learning approach to extract and document the guideline changes that I observed. Inductive learning allows people to discover patterns in a seemingly chaotic collection of observations [273] making it a powerful strategy for helping people deepen their understanding of content [274]. Following the inductive learning approach, I begin by selecting a candidate guideline recommendation from the GEM encoded sample. Thereafter, I analysed the candidate recommendation to identify its equivalent guideline recommendation in a successive version of the CPG. In those instances where an equivalent guideline recommendation was found, I compared the two versions of the candidate guideline recommendation. For those guideline recommendation sets that were found to be different, I recorded the change by specifying the change or refining an existing specification of the change. I repeated this process until all the candidate guideline recommendations were analysed in each comparison cycle. Refer to figure 4.1 for an overview of the comparison cycle within the change specification process.

Finally, I analysed the recorded incidents of the changes between the first and the second version of the CPGs. I repeated this analysis with the second and the third version of the CPGs.

4.3 Types of CPG changes and their incidence

4.3.1 Encoding clinical practice guideline recommendations

Firstly, I obtained several copies of CPG documents from the Malawi the Ministry of Health. I identified three successive versions of CPGs from the copies of the CPG documents. The identified CPGs were used by health workers for the provision of HIV services. The three versions of the CPGs were published by the Malawi Ministry of Health in 2008, 2011 and 2014.

Secondly, I examined 114 pages of the 2008 version of the CPG text for managing HIV services for the presence of candidate guideline recommendations within the CPGs. I identified 13 CPGs from which I extracted and encoded 105 candidate guideline recommendations into GEM.

Thirdly, I examined 79 pages of the 2011 version of the CPG text for managing HIV services for the presence of candidate guideline recommendations within the CPGs. I

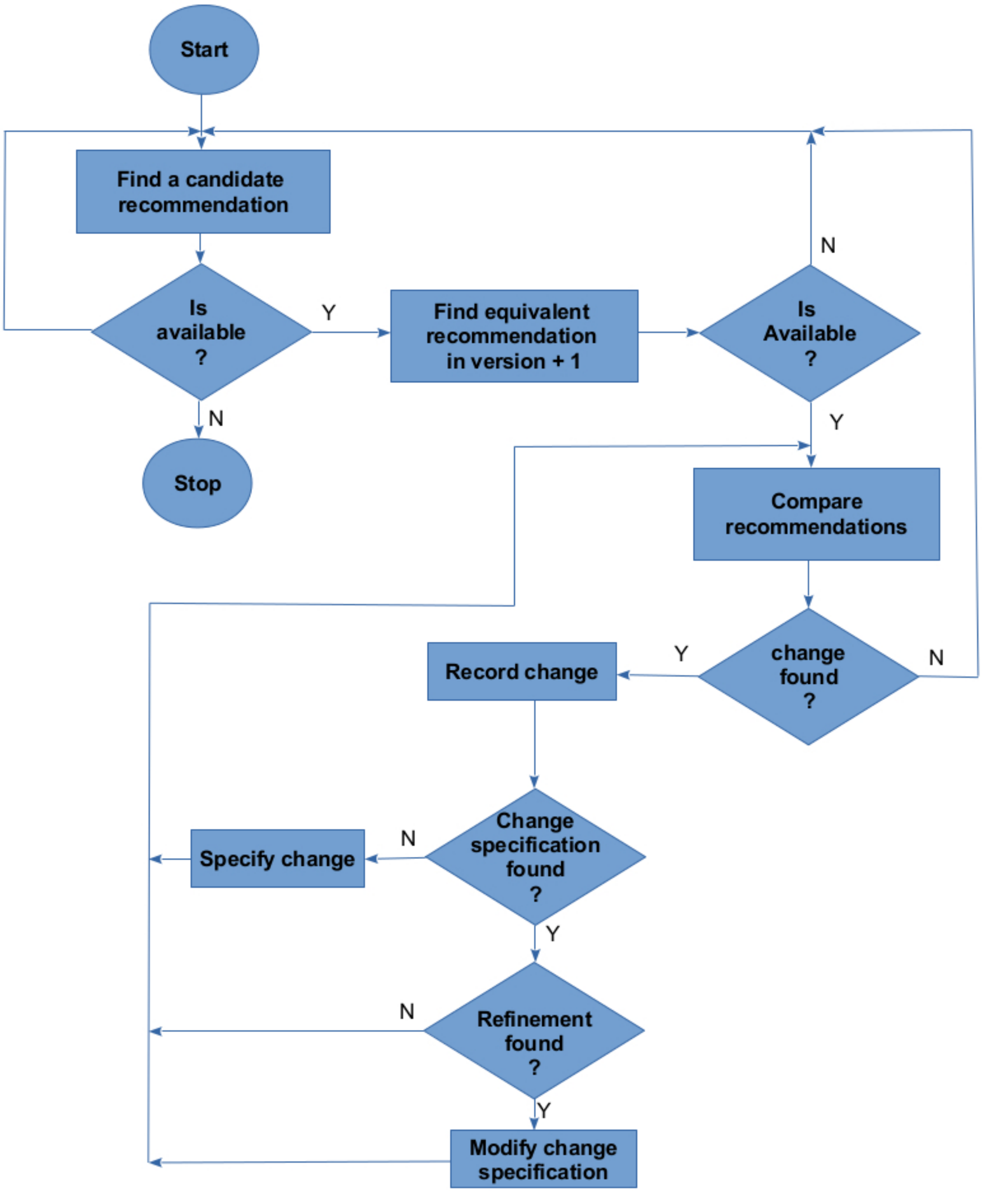

Start
Find a candidate recommendation
Is available ?
Y
N
Stop
Find equivalent recommendation in version + 1
Is Available ?
N
Y
Compare recommendations
change found ?
Y
N
Record change
Change specification found ?
N
Y
Specify change
Refinement found ?
N
Y
Modify change specification

FIGURE 4.1: Change specification process

identified 21 CPGs from which I extracted and encoded 85 candidate guideline recommendations into GEM.

Finally, I repeated the previous process with the 2014 version of the CPG text. I examined 95 pages of the guideline text for the presence of candidate guideline recommendations within the CPGs. I identified 19 CPGs from which I extracted and encoded 98 candidate guideline recommendations into GEM. Refer to appendix A for detailed report extracts of the 2008, 2011 and 2014 versions of GEM encoded CPGs for the *"Malawi Clinical Management of HIV"*. For a summary of the totals of the GEM encoded guidelines, refer to table 4.1.

TABLE 4.1: Summary of the characteristics of the GEM encoded CPGs

Year	**No. of pages**	**No. of CPGs**	**No. of recommendations**
2008	114	13	105
2011	79	21	85
2014	95	19	98
Totals	**288**	**53**	**288**

4.3.2 Comparing CPG versions

To begin with, I compared the 2008 with the 2011 version of the GEM encoded CPGs to identify and specify the changes between the two. After analysing the 105 and 85 candidate guideline recommendations from the 2008 and 2011 CPG versions respectively, I noted that 75 candidate guideline recommendations from the 2008 CPG version were discontinued in the 2011 CPG version. Of the 30 candidate guideline recommendations that were carried over from the 2008 CPG version to the 2011 CPG version, I noted that 27 were changed with 41 individual changes in total. Furthermore, I noted that 55 new candidate guideline recommendations were introduced in the 2011 GEM encoded sample.

Thereafter, I repeated the process with the 2011 and the 2014 versions of the GEM encoded CPGs. I compared the candidate guideline recommendations in the two CPG versions to identify and specify changes between the two. After analysing the 85 and 98 guideline recommendations from the 2011 and 2014 CPG versions respectively, I noted that three candidate guideline recommendations from the 2011 CPG version were discontinued in the 2014 CPG version. I also noted that, of the 82 candidate guideline recommendations that were carried over from the 2011 CPG version to the 2014 CPG version, 12 were changed with 17 individual changes in total. Furthermore, I noted that 16 new candidate guideline recommendations were introduced in the 2014 GEM encoded sample. Table 4.2 shows a summary of the characteristics of the guideline changes that I identified. The raw data regarding the CPG change incidents is available in Appendix J.

In the next section, I will categorise and describe the changes that I have identified in this section using both an example of the actual changes—indicated in bold in the examples—and a generalised specification of the change.

For this generalised specification, I use the following notation. Let the following denotations hold for CPG structural elements: clinical practice guideline be CPG, guideline recommendation be GR, condition be C, decision variable be DV, value for a decision variable be DVV, recommended action be RA, action verb be AV, and a verb complement for an action verb be AVC. Then the state before the change is indicated with a subscript time “$_t$” and the changed CPG one chronon later, “$_{t+1}$” (which for the examined CPGs is 3 years). Anything unchanged is omitted from the notation to avoid clutter.

Table 4.2: Characteristics of CPG changes

Characteristic	**2008/2011**	**2011/2014**
No. of discontinued recommendations	75	3
No. of new recommendations	55	16
Instances of changes to recommendations	41	17
No. of recommendations with changes	*27*	*12*
Total number of changes in CPGs	**171**	**36**

4.3.3 CPG Change Categories

There are 10 categories in total. The changes are discussed in the following sections.

4.3.3.1 Addition of a decision variable to a guideline condition

A decision variable can be added to the condition for CPG execution if and only if the condition for the guideline recommendation already has at least one decision variable.

$\mathsf{CPG}_t = \{\mathsf{DV}_1, ..., \mathsf{DV}_n; ...\}$

$\mathsf{CPG}_{t+1} = \{\mathsf{DV}_1, ..., \mathsf{DV}_{n+1}; ...\}$

For example, the guideline recommendation for providing ART in special situations for older children from age three years and older that have active TB, had an additional decision variable *‘weight’* introduced in the 2014 version of the CPG as shown below:

2011 version

```
IF Active TB = [present] AND
   On ART = [No] AND
   Age >= [3 years & < 15 years] AND
   HIV test result = [Positive]
THEN
   Prescribe AZT/3TC + EFV AND
   Initiate ART within 14 days
```

2014 version

```
IF Active TB = [present] AND
   On ART = [No] AND
   HIV test result = [positive] AND
   Age >= [3 years] AND
   Weight < [35 kg]
THEN
   Prescribe regimen 4P/4A AND
   Initiate ART within 14 days
```

4.3.3.2 Change of a decision variable value

A value of a decision variable can be changed in a condition for CPG execution when a guideline recommendation has a set of decision variables for a condition under which the guideline is to execute when the condition is true.

$\mathsf{CPG}_t = \{\mathsf{C}_1\{\mathsf{DV}_1,\mathsf{DVV}_1=\mathsf{a}\}, ..., \ \mathsf{C}_n\{\mathsf{DV}_n,\mathsf{DVV}_n\}; ...\}$
$\mathsf{CPG}_{t+1} = \mathsf{C}_1\{\{\mathsf{DV}_1,\mathsf{DVV}_1=\mathsf{b}\}, ..., \ \mathsf{C}_n\{\mathsf{DV}_n,\mathsf{DVV}_n\}; ...\}$
where $\mathsf{a} \neq \mathsf{b}$
Equally, the guideline recommendation for determining ART eligibility in children aged 12 months and above, had the value of the decision variable *Age* changed as follows: For this generalised specification, I use the following notation. Let the following denotations hold for CPG structural elements: clinical practice guideline be CPG, guideline recommendation be GR, condition be C, decision variable be DV, value for a decision variable be DVV, recommended action be RA, action verb be AV, and a verb complement for an action verb be AVC. Then the state before the change is indicated with a subscript time "$_t$" and the changed CPG one chronon later, "$_{t+1}$" (which for the examined CPGs is 3 years). *2011 version*

```
IF Age = [12 to 24 months] AND
   HIV test result = [positive]
THEN
   Patient is eligible for ART
```

2014 version

```
IF Age = [12 to 60 months] AND
   HIV test result = [positive]
THEN
   Patient is eligible for ART
```

4.3.3.3 Removal of a decision variable from a guideline condition

A decision variable can be removed from the condition for CPG execution if and only if the set of decision variables under which the guideline is to execute when the condition is true, has at least two variables.

$\mathsf{CPG}_t = \{\mathsf{DV}_1 \ldots \mathsf{DV}_n; \ldots\}$
$\mathsf{CPG}_{t+1} = \{\mathsf{DV}_1 \ldots \mathsf{DV}_{n-1}; \ldots\}$
For instance, the guideline recommendation for suspecting ART failure due to a new WHO clinical stage condition, had the *CD4 count* decision variable removed from the 2011 version of the CPG as follows:

2008 version

```
IF On ART = [Yes] AND
   Duration of ART = [12 months or more] AND
   Adhering to therapy = [Yes] AND
   New WHO stage 4 = [Present] AND
   CD4 count = [< 200 cells/mm$^3$]
THEN
   Patient has suspected ART drug failure
```

2011 version

```
IF On ART = [Yes] AND
   Duration of ART = [12 months or more] AND
   New WHO stage 3 or 4 = [Present] AND
THEN
   Patient has suspected ART drug failure
```

4.3.3.4 Change of a decision variable

A decision variable can be replaced with another decision variable within a condition for CPG execution if and only if the condition under which the guideline is to execute exists.

$\mathsf{CPG}_t = \{\{\mathsf{DV}_1\mathsf{=a}\} \ldots \mathsf{DV}_n; \ldots\}$
$\mathsf{CPG}_{t+1} = \{\{\mathsf{DV}_1\mathsf{=b}\} \ldots \mathsf{DV}_n; \ldots\}$
where $\mathsf{a} \neq \mathsf{b}$
For example, the guideline recommendation for determining dosage of CPT in infants aged five months or younger, had the *Age* decision variable changed to *weight* in the inclusion criteria of the guideline recommendation for the 2011 version of the CPG as follows:

2008 version

```
IF Age = [6 weeks to 5 months] AND
   Eligible for CPT = [Yes]
```

```
THEN
   CPT dosage is 120mg once a day
```

2011 version

```
IF Weight = [< 6kg] AND
   Eligible for CPT = [Yes]
THEN
   CPT dosage is 120mg once a day
```

4.3.3.5 Addition of a recommended action

A recommended action can be added to a set of recommended actions within a guideline recommendation for CPG execution if and only if a guideline recommendation has got at least one recommended action.

$\mathsf{CPG}_t = \{\mathsf{RA}_1 \; ...\mathsf{RA}_n; \; ...\}$

$\mathsf{CPG}_{t+1} = \{\mathsf{RA}_1 \; ...\mathsf{RA}_{n+1}; \; ...\}$

Equally, the guideline recommendation for managing a patient presenting with renal failure, had an additional recommended action added to the 2011 version as follows:

2008 version

```
IF On ART = [No] AND
   Renal failure = [Present] AND
   HIV test result = [Positive]
THEN
   Refer to district or central hospital AND
   Do not exclude from treatment
```

2011 version

```
IF On ART = [No] AND
   Renal failure = [Present] AND
   Age category = [Adult] AND
   HIV test result = [Positive]
THEN
   Refer to district or central hospital AND
   Prescribe Regimen 4 AND
   Start ART within 7 days of diagnosis
```

4.3.3.6 Removal of a recommended action

A recommended action can be removed from an action set for CPG execution if and only if the guideline recommendation under which the CPG is to execute has more than one recommended action.

$\mathsf{CPG}_t = \{\mathsf{RA}_1 \ \ldots\mathsf{RA}_n; \ \ldots\}$

$\mathsf{CPG}_{t+1} = \{\mathsf{RA}_1 \ \ldots\mathsf{RA}_{n-1}; \ \ldots\}$

For example, the guideline recommendation for managing a patient presenting with renal failure, had the recommended action for "automatically excluding patients from treatment" removed from the 2011 version of the CPG as follows:

2008 version

```
IF On ART = [No] AND
   Renal failure = [Present] AND
   HIV test result = [Positive]
THEN
   Refer to district or central hospital AND
   Do not exclude from treatment
```

2011 version

```
IF On ART = [No] AND
   Renal failure = [Present] AND
   Age category = [Adult] AND
   HIV test result = [Positive]
THEN
   Refer to district or central hospital AND
   Prescribe Regimen 4 to patient AND
   Start ART within 7 days of diagnosis
```

4.3.3.7 Change of an action verb complement

A verb complement of an action verb within a recommended action for CPG execution, can be replaced with another verb complement, if and only if at least one recommended action in the guideline recommendation under which the CPG is to execute exists.

$\mathsf{CPG}_t = \{\mathsf{RA}_1\{\mathsf{AV}_1,\mathsf{AVC}_1{=}\mathsf{a}\}, \ \ldots, \ \mathsf{RA}_n\{\mathsf{AV}_n,\mathsf{AVC}_n\}; \ \ldots\}$

$\mathsf{CPG}_{t+1} = \{\mathsf{RA}_1\{\mathsf{AV}_1,\mathsf{AVC}_1{=}\mathsf{b}\}, \ \ldots, \ \mathsf{RA}_n\{\mathsf{AV}_n,\mathsf{AVC}_n\}; \ \ldots\}$

where $\mathsf{a} \neq \mathsf{b}$

For instance, the guideline recommendation for managing a patient presenting with renal failure in adults, had the action verb complement for the ‘medication prescription’ action changed in the 2014 version of the CPG as follows:

2011 version

```
IF On ART = [No] AND
   Renal failure = [Present] AND
   HIV test result = [Positive] AND
   Age = [>= 15 years]
THEN
   Refer to district or central hospital AND
   Prescribe Regimen 4 AND
   Start ART within 7 days of diagnosis
```

2014 version

```
IF On ART = [No] AND
   Renal failure = [Present] AND
   HIV test result = [Positive]
THEN
   Refer to district or central hospital AND
   Prescribe Regimen 0 AND
   Start ART within 7 days of diagnosis
```

4.3.3.8 Change of a recommended action

A recommended action for CPG execution can be replaced with another recommended action if and only if the guideline recommendation under which the CPG is to execute has at least one recommended action.

$\mathsf{CPG}_t = \{\{\mathsf{RA}_1\mathsf{=a}\}\ \ldots\mathsf{RA}_n;\ \ldots\}$
$\mathsf{CPG}_{t+1} = \{\{\mathsf{RA}_1\mathsf{=b}\}\ \ldots\mathsf{RA}_n;\ \ldots\}$
where $\mathsf{a} \neq \mathsf{b}$
For example, the guideline recommendation for managing a patient presenting with acute hepatitis, had the recommended action changed in the 2011 version of the CPG as follows:

2008 version

```
IF On ART = [No] AND
   Jaundice = [Present] AND
   HIV test result = [Positive]
```

```
THEN
   Do not prescribe Regimen 1
```

2011 version

```
IF On ART = [No] AND
   Jaundice = [Present] AND
   HIV test result = [Positive]
THEN
   Refer to district/central hospital
```

4.3.3.9 Addition of a guideline recommendation

A candidate guideline recommendation can be added to a CPG if and only if the CPG has at least one existing candidate guideline recommendation.

$\mathsf{CPG}_t = \{\mathsf{GR}_1 \ldots \mathsf{GR}_n; \ldots\}$

$\mathsf{CPG}_{t+1} = \{\mathsf{GR}_1 \ldots \mathsf{GR}_{n+1}; \ldots\}$

For example, the candidate guideline recommendation for determining ART eligibility in infants presenting with Cryptococcal meningitis was added to the 2011 version of the CPG for determining ART eligibility.

4.3.3.10 Removal of a guideline recommendation

A candidate guideline recommendation can be removed from a CPG if and only if the CPG has two or more candidate guideline recommendations.

$\mathsf{CPG}_t = \{\mathsf{GR}_1 \ldots \mathsf{GR}_n; \ldots\}$

$\mathsf{CPG}_{t+1} = \{\mathsf{GR}_1 \ldots \mathsf{GR}_{n-1}; \ldots\}$

For instance, the candidate guideline recommendation for determining ART eligibility in infants associated with recent HIV related maternal death was removed from the 2011 version of the CPG for determining ART eligibility.

4.3.4 CPG change occurrences

Given the aforementioned type of changes, we analysed their incidence for the HIV CPGs of Malawi. A summary of the frequencies of CPG changes is presented in Table 4.3.

Of the 30 candidate guideline recommendations that were carried over from the 2008 to the 2011 version of the CPGs, 27 were modified with 41 incidents of guideline recommendation changes. Similarly, 12 candidate guideline recommendations were modified with 17 incidents of guideline recommendation changes from the 82 candidate guideline recommendations that were carried over from the 2011 to the 2014 version of the CPGs.

The frequencies of the changes that were identified between 2008 and 2011 version of the CPGs were higher than those frequencies identified between the 2011 and the 2014 version of the CPGs. Three types of changes, addition of a a recommended action, removal of a recommended action and change of recommended action, that occurred between the 2008 and 2011 version of the CPGs were not identified in the changes that occurred between the 2011 and the 2014 version of the CPGs.

TABLE 4.3: Frequency of changes in the Malawi clinical management of HIV CPGs.

Category	**Type of change**	**2008/2011**		**2011/2014**	
		#	***% of total***	**#**	***% of total***
Decision	Addition of a decision variable to a guideline condition	3	1.8	2	5.6
	Change of a decision variable value	13	7.6	4	11.1
	Removal of a decision variable from a guideline condition	2	3.8	2	1.2
	Change of a decision variable	12	7.0	5	13.9
Action	Addition of a recommended action	2	1.2	0	0.0
	Removal of a recommended action	2	1.2	0	0.0
	Change of an action verb complement	5	2.9	4	11.1
	Change of a recommended action	2	1.2	0	0.0
Recommendation	Addition of a recommendation	55	32.2	16	44.4
	Removal of a recommendation	75	43.9	3	8.3

4.3.5 Analysis of existing CIG models

Having identified the characteristics of CPG changes, we further analysed existing CIG models with respect to the changes. CIG models were identified from existing literature accessible through Google Scholar, PubMed and Open Clinical portal. GLIF, Arden Syntax, SAGE, EON, PROforma and Asbru [11] were analysed with regards to CPG changes. Project websites, where accessible, were accessed to analyse any tutorials and demonstrations that were found. Any knowledge modelling tools that were freely available were also downloaded and analysed. We analysed Arden Syntax Checker for Windows, AsbruView, Tallis, SAGE workbench and GLIF Ontology in Protégé.

The results of the analysis are now presented. Table 4.4 summarises CPG structural components that are explicitly defined in existing CIG models and are directly accessible within existing knowledge modelling tools. An alignment of our terms and representation primitives from the existing CIG models are presented in table 4.5. From these results, we can see that fine-grained CPG structural components that are affected by the changes characterised in the previous section, are not explicitly defined as modelling elements in existing CIG models. The fine-grained components in existing CIG models are implicitly defined as part of other structural components, specified either as free text or using a formal expression language such as GELLO [96]. As a result, all the changes that were presented in the previous section and affect fine-grained CPG structural components cannot be represented using concepts from existing CIG models.

Table 4.4: CPG structural components explicitly defined in CIG models.

Structural Component	**Arden Syntax**	**GLIF**	**SAGE**	**EON**	**PROforma**	**Asbru**
Condition (C)	Yes	Yes	Yes	Yes	Yes	Yes
Decision variable (DV)	No	No	No	No	No	No
Decision variable value (DVV)	No	No	No	No	No	No
Recommended action (RA)	Yes	Yes	Yes	Yes	Yes	Yes
Action verb (AV)	No	No	No	No	No	No
Action verb complement (AVC)	No	No	No	No	No	No

Table 4.5: Representation primitives in existing CIG models.

Structural Component	**Arden Syntax**	**GLIF**	**SAGE**	**EON**	**PROforma**	**Asbru**
Condition (C)	Logic slot	Decision step	Decision	Decision	Decision	Plan
Recommended action (RA)	Action slot	Action spec	Action	Action/ Activity	Action/ Inquiry	Condition

4.4 Discussion

Evidence-based CPGs, available to task-shifted health workers at the point-of-care through CDSSs, are a necessity towards improving the delivery of care. Due to the evolving nature of clinical practice, CIGs in CDSSs need to be updated on a continuous basis to provide recommendations based on up-to-date evidence-based CPGs. Characteristics of

CPG changes need to be specified precisely if CPG change operations are to be modelled accurately to ensure adequate clinical knowledge evolution support in CIGs. The complexity of CIG models, a demanding and time-consuming formalisation process, demands automating parts of the modelling process and modelling adaptations to decrease the required implementation effort for CIGs [61].

Fundamental change operations of a CIG are defined from the addition, removing and changing of basic structural components of a CPG. The 10 types of CPG changes identified within three categories cover changes that affect both coarse-grained and fine-grained structural components of a CPG. The coarseness of the basic structural components in existing CIG models can not sufficiently allow for specification of precise semantics for CPG change operations. This limits the ability to explore the properties and limitations of existing CIG models with respect to handling the 10 types of CPG changes. Extensions to include the fine-grained structural components in a CIG would provide a sufficient foundation for modelling change operations that can be used to provide adequate knowledge evolution support in an implementation.

The categorisation of changes can also apply to other developing countries with similar characteristics, such as those in the sub-Saharan Africa. In addition, the categorisation of changes can also apply to other clinical domains that deploy or task-shifted health workers such as clinical management of diabetes.

There was a significant difference between the updates introduced in the 2011 and the 2014 version of the CPGs with regards to the number of new guideline recommendations that were introduced in each update. This might have been due to the fact that Malawi introduced new recommendations into the 2011 Malawi CPGs for providing HIV services based on the 2010 World Health Organisation (WHO) recommendations for managing HIV-positive pregnant and breastfeeding women [275].

Having identified the characteristics of types of CPG changes and their incidences, precise semantics for modelling and handling CPG changes can be formally specified. The CPG changes and their related semantics would give one a basis for modifying and extending existing CIG models and related modelling platforms with capabilities for handling CPG updates.

4.5 Chapter summary

I have described an evidence-based characterisation of the types of CPG changes and their incidences. For each type of change, I have specified precisely each structural component that exhibits a change and a description of how that particular change occurs to facilitate its application. The characterisation of the types of changes serve as a foundation for devising formal semantics that can support the handling of CPG updates in a CIG model suitable for regions that have adopted task-shifting. A CIG conceptual model that is aware of the elements that are affected by CPG changes can serve as a template for providing tooling that can support evolving CIGs in a model-driven clinical information system.

In the next chapter, I conceptualise an evolving CIG and propose a layered architecture for modelling these evolving CIGs. I further evaluate the novel CIG modelling architecture for its adequacy in modelling an evolving CIG and its ability to enable smart-editing features for the CIG elements that are affected CPG changes.

Chapter 5

Modelling evolving computer-interpretable guidelines

5.1 Introduction

This chapter introduces the concepts of evolving CIG and evolving CPG formalization. I start by discussing how a novel CIG modelling architecture was developed following a model-driven approach. I further discuss how I evaluated this novel architecture for its adequacy in supporting the authoring and maintenance of CIGs.

5.2 Evolving clinical practice guideline formalization

An electronic medical record system that uses CPGs should allow changes to be applied to its CIGs systematically when changes to CPGs occur. In order to provide automated or semi-automated support to the CIG modeller when applying these changes, an evolving CIG conceptual model is required. The specification of the evolving CIG conceptual model should contain explicit and invariant definitions of all the elements that are affected by CPG changes. Recall the types of CPG changes summarised in Table 5.1. A language model and a concrete syntax can then be mapped from this evolving CIG conceptual model, enabling the automated creation of tools that can support the authoring and maintenance of CIGs through smart-editing features.

An individual CPG has one or more guideline recommendations. Each of the guideline recommendations has one or more conditions, which must be satisfied in order for a particular recommended action to be executed. Each condition has a decision variable that is tested to determine whether it is appropriate to execute one or more related

TABLE 5.1: Types of CPG changes [2]

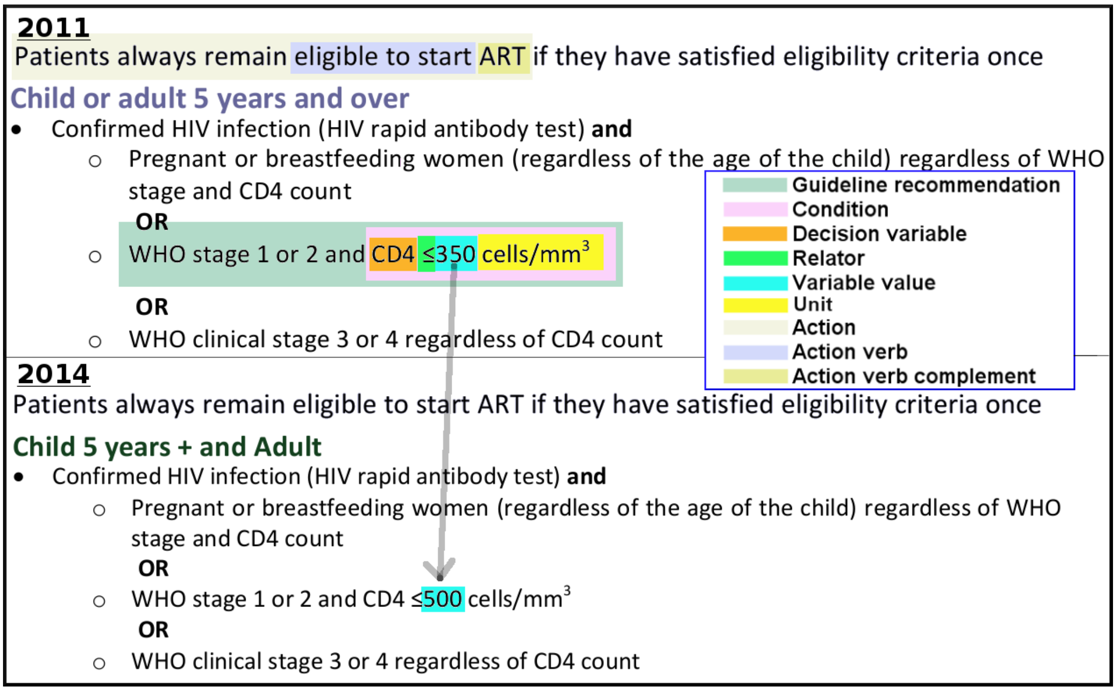
Category	**Type of change**
Updating decisions	Adding a decision variable
	Changing a value of a decision variable
	Removing a decision variable
	Changing a decision variable
Updating actions	Adding a recommended action
	Removing a recommended action
	Changing an action verb complement
	Changing a recommended action
Updating guideline recommendations	Adding a guideline recommendation
	Removing a guideline recommendation

recommended actions. The actual value of a decision variable that is tested whether a particular set of recommended actions is appropriate or not, is referred to as a variable value. Different variable values can be stored in different units of measure. Each recommended action within the guideline has an action verb that determines the type of action to be taken, and an action verb complement that completes the sense of the action verb by referring to a direct or indirect object. When a new version of a CPG is introduced, any of these concepts can be affected.

Let me illustrate this with an example, so as to clarify the context in which evolving CIGs are used. I use the CPG for determining ART eligibility in children and adults aged five years and above from the Malawi guidelines for the clinical management of HIV of 2011 and 2014. Extracts of the CPG from the guideline documents of 2011 and 2014 are presented in Fig. 5.1. The elements that make up the CPG are highlighted and the change has been explicitly pointed out in the figure: when a new version of the CPG was introduced in 2014, a change of type 'changing a type of a decision variable' had occurred.

5.3 Materials and methods

In order to evaluate the feasibility of creating a modelling architecture that can support evolving CPG formalization, I started by defining a four-layer CIG modelling architecture that is based on a model-driven engineering approach. Thereafter, I defined a formal evolving CIG conceptual model that explicitly specifies the elements that are affected when CPG changes occur. I later systematically map the evolving CIG conceptual model into a comprehensive language model that includes denotational semantics and a formal abstract syntax. I mapped the formal language model into a sufficient concrete syntax specification for a model-based DSL that is tailored towards the modelling of evolving



2011

Patients always remain eligible to start ART if they have satisfied eligibility criteria once

Child or adult 5 years and over

- Confirmed HIV infection (HIV rapid antibody test) **and**
  - ○ Pregnant or breastfeeding women (regardless of the age of the child) regardless of WHO stage and CD4 count
    OR
  - ○ WHO stage 1 or 2 and CD4 ≤350 cells/mm^3
    OR
  - ○ WHO clinical stage 3 or 4 regardless of CD4 count

2014

Patients always remain eligible to start ART if they have satisfied eligibility criteria once

Child 5 years + and Adult

- Confirmed HIV infection (HIV rapid antibody test) **and**
  - ○ Pregnant or breastfeeding women (regardless of the age of the child) regardless of WHO stage and CD4 count
    OR
  - ○ WHO stage 1 or 2 and CD4 ≤500 cells/mm^3
    OR
  - ○ WHO clinical stage 3 or 4 regardless of CD4 count

🟩 Guideline recommendation
🟪 Condition
🟧 Decision variable
🟩 Relator
🟦 Variable value
🟨 Unit
⬜ Action
🟦 Action verb
🟨 Action verb complement

FIGURE 5.1: CPG for determining antiretroviral therapy (ART) eligibility in 2011 and 2014 of Malawi HIV CPGs, with each fine-grained element colour coded (see legend on the right of the figure as insert)

CIGs, that I have named *FCIG*. Finally, I carried out a scenario-based and an empirical evaluation of *FCIG* using CPGs from Malawi.

5.3.1 Research questions

Recall the second research question **RQ2** in Section 3.1:

Can a model-driven engineering approach adequately support the modelling of an evolving CIG?

In this study, I split the evaluation in two dimensions to address research question **RQ2**. The first dimension focused on CPG representation adequacy. The second dimension focused on the capacity to support the CIG modeller when applying a specific change. As a result, I formulated two subquestions to guide the evaluations for each of the two dimensions as follows:

RQ2-S1: *Can **FCIG** be used to model an evolving CIG adequately?*

RQ2-S2: *Can **FCIG** directly support the application of fine-grained CPG changes?*

I investigated the representational accuracy and adequacy of *FCIG* including *FCIG*'s ability to provide smart-editing support for the CPG elements that are affected by a change. I obtained CPG documents from Malawi and encoded guideline recommendations from the CPG documents using *FCIG*. To answer sub-question **RQ2-S1**, I measured the number of guideline recommendations that were encoded using *FCIG* successfully. To answer **RQ2-S2**, I measured the number of language concepts that were directly supported by smart-editing features when a language concept was affected by a CPG change.

5.3.2 Research approach

I used a quantitative approach to conduct the two studies for this evaluation. The first, was to assess the adequacy in encoding an evolving CIG using *FCIG*. The second, was to assess *FCIG*'s ability to support, and awareness of, evolving CIG concepts that are affected when a change occurs. I collected quantitative data regarding the number of guidelines and associated guideline recommendations that were able to satisfy each of the two requirements.

5.3.3 Study design

I obtained a convenient sample of CPG documents from the Malawi Ministry of Health. The CPG documents were the ones that the officials from the Central Monitoring and Evaluation Division (CMED) of the Malawi Ministry of Health had provided. From the convenient sample of CPG documents, I selected the Malawi integrated guidelines for the clinical management of HIV and the guideline for community integrated management of child illnesses (IMCI) because they are comprehensively documented and are operationalised within the national EMR systems in Malawi. I further used the stratified random sampling technique to select representative samples of guideline recommendations from the integrated guidelines. Stratified random sampling allows a researcher to obtain a sample that best represents an entire population under study by ensuring the presence of key subgroups within a population [231, 233]. I used the Malawi medical concept dictionary to obtain a controlled set of medical vocabulary.

5.3.3.1 Evaluation study one: Assessing *FCIG*'s CPG representation adequacy

In this study, I set out to answer research subquestion RQ2-S1. Noting that the structure of guideline recommendations are usually similar within a particular guideline, I stratified the guideline recommendations within the integrated guidelines by guideline. From each guideline, I selected a weighted random sample of guideline recommendations for inclusion. I derived the randomised sample through a digital randomisation application called Random UX[1]. Thereafter, I encoded the selected guidelines using *FCIG*.

5.3.3.2 Evaluation study two: Assessing *FCIG*'s ability to support CPG changes

In this study, I set out to answer research subquestion RQ2-S2. Noting that the operations required to apply a particular type of change are similar, I stratified the guideline recommendations from the Malawi HIV guidelines by type of CPG change. For each type of CPG change, I selected a weighted random sample of guideline recommendations from the 2008 version of the guidelines. I derived the randomised sample through a digital randomisation application called Random UX[1]. I later encoded each selected guideline recommendation using *FCIG*. For each encoded guideline recommendation, I applied its associated change that was required to update it to its 2011 version.

5.3.4 Criteria to address the research question RQ2

In this section, I describe the criteria that I used to address the second research question **RQ2** through subquestions **RQ2-S1** and **RQ2-S2**.

5.3.4.1 Criteria to address research subquestion RQ2-S1

In order to address research question **RQ2-S1**, I recorded the total number of guideline recommendations in the guideline strata and attempted to model the guideline recommendations using *FCIG*. Recall, from the discussions in Chapter 4, that the conceptual models of existing CIG modelling languages do not have explicit concepts for specifying fine-grained CPG representation primitives such as action verb and verb complement. Hence, adequate representation of a recommendation in a CPG was regarded as an instance whereby *FCIG* modelling primitives were used to specify all guideline recommendation concepts, including fine-grained constructs. I recorded the number of guideline recommendations from the guideline strata that were adequately encoded using *FCIG* in order to measure task completion rate.

I formulated the following hypothesis to guide my analyses:

H_0: *FCIG* can not be used to complete CPG encoding tasks.

H_1: *FCIG* can be used to complete CPG encoding tasks adequately.

[1] https://play.google.com/store/apps/details?id=ru.uxapps.random&hl=en

5.3.4.2 Criteria to address research subquestion RQ2-S2

In order to address research question **RQ2-S2**, I encoded the 2008 versions of the guideline recommendations from the CPG change type strata using *FCIG*. I recorded the total number of required CPG changes in the CPG change type strata. Thereafter, I attempted to apply the individual changes to the encoded guidelines. I took note whether the evolving CIG semantic elements that were affected by each change were directly supported by smart-editing features. I tracked the number of individual CPG change occurrences whose affected elements were directly supported with smart-editing features. This allowed me to measure task completion rate.

I formulated the following hypothesis to guide my analyses:

$\mathbf{H}_0$: *FCIG* can not provide smart-editing support for CPG maintenance tasks.

$\mathbf{H}_1$: *FCIG* can provide smart-editing support for CPG maintenance tasks.

5.4 A four-layer architecture for modelling evolving CIGs

Model-driven engineering can be based on a four-layer modelling architecture [276–278], which has been extended to cater for its interaction with Semantic Web technologies, notably the logic-based ontologies specified in the OWL languages and automated reasoning over the MDE's metamodels (OWL's so-called TBox) and models (OWL's so-called ABox) [277]. Staab et al.'s extension [277] was refined for to suit CIG change management, which is presented in Fig. 5.2.

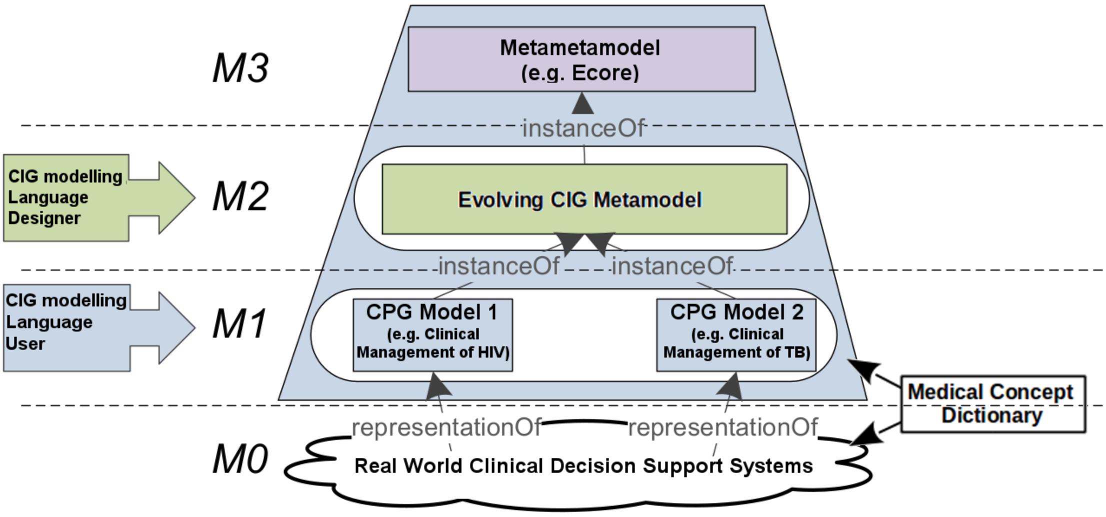

M3
Metametamodel (e.g. Ecore)
instanceOf
M2
CIG modelling Language Designer
Evolving CIG Metamodel
instanceOf
instanceOf
M1
CIG modelling Language User
CPG Model 1 (e.g. Clinical Management of HIV)
CPG Model 2 (e.g. Clinical Management of TB)
Medical Concept Dictionary
representationOf
representationOf
M0
Real World Clinical Decision Support Systems

FIGURE 5.2: Four-layer CIG modelling architecture

Layers M0 and M1 are concerned with the definitions of specific CIG models and layers M2 and M3 of the CIG modelling architecture are mainly concerned with CIG modelling language definition. More precisely, layer M0 is the lowest level of CIG abstraction, which comprises CIGs in real world clinical decision support systems. Layer M1 is an abstraction of CIGs at layer M0. At this layer, domain-specific CIG models are specified such that the CIGs at layer M0 conform to their CIG models at layer M1. Layer M2 is an abstraction of the CIG models at layer M1 such that the CIG models at layer M1 conform to the CIG metamodel at layer M2. Layer M3, in turn, is an abstraction of layer M2 such that the CIG metamodel at layer M2 conforms to the metametamodel at layer M3. This layer can be handled by the usual metametamodel languages (OWL's metamodel or Ecore in praxis).

5.5 Evolving computer-interpretable guideline conceptual model

I present a conceptual model of an evolving CIG in a formal (logic-based) specification. The evolving CIG conceptual model is mapped into an evolving CIG metamodel at layer M2 of the CIG modelling architecture of Fig. 5.2. The resultant UML Class Diagram of the evolving CIG conceptual model showing an informal overview is presented in Fig. 5.3.

5.5.1 Conceptual model development: the process

The development of the metamodel was informed by both existing CIG modelling proposals and by obtaining modelling guidelines from foundational ontologies.

First, the metamodel builds on existing CIG modelling formalisms and refines it through explicitly specifying the fine-grained CPG components affected by CPG changes. GEM, GLIF, Asbru, PROforma, SAGE and Arden syntax were evaluated to identify guideline modelling primitives. The CIG metamodel was iteratively constructed by analysing the characteristics of each type of CPG change and subsequently including all elements that are affected by that particular change to the CIG metamodel. Decisions and actions were found to be the guideline representation primitives in existing guideline modelling languages, concurring with [2], and lacking vocabulary for elements such as decision variable values that can change (recall Fig. 5.1: a value changing from ≤ 350 to ≤ 500). Thus, such entities were incorporated into the CIG metamodel.

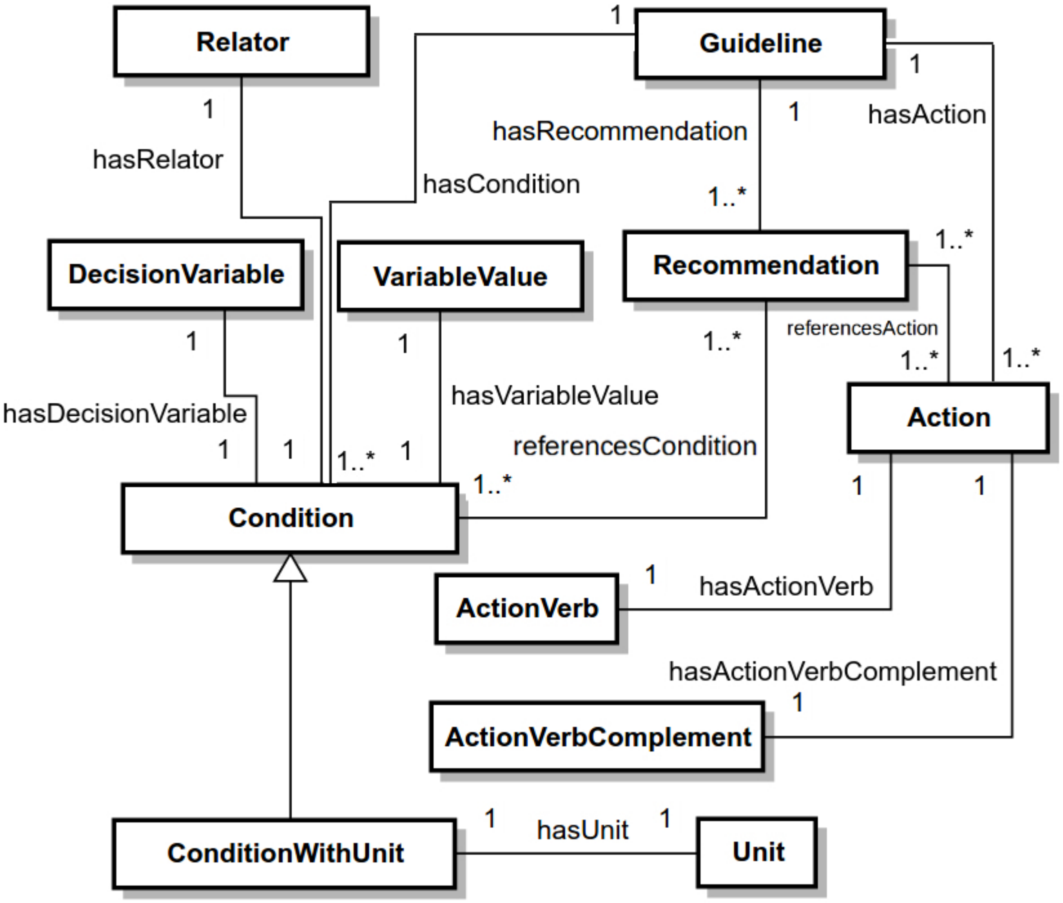


FIGURE 5.3: UML class diagram of an evolving CIG

Second, with as aim to foster wide interoperability between CIG formalization systems, the CIG metamodel was extended by availing the knowledge from the Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT), which is an ontology that is one of the most comprehensive clinical terminology systems [279] and relatively widely used for electronic health records. In particular, the concept *Decision Variable* now refers to the same concept as *Clinical Finding* in SNOMED CT and the CIG metamodel was further extended by the concept *Unit* in SNOMED CT.

Third, noting that also SNOMED CT has been analysed with respect to the DOLCE foundational ontology [280], I considered foundational ontologies for further quality improvements regarding the guideline modelling primitives. As a full linking to any of the foundational ontologies would introduce many new terms that are irrelevant for a practical domain-specific modelling language, I decided to only borrow ideas from it. This resulted in an extension of the metamodel with the concept *Relator* from the General Formal Ontology (GFO), which is a foundational ontology for integrating objects and processes [281].

5.5.2 Precise specification of the evolving CIG conceptual model

This section specifies a precise semantics for the fine-grained CIG conceptual model presented in Section 5.5.1. The semantics presented in this section were expressed in OWL using the Protégé [282] ontology development environment. I use Berardi et al's [283] encoding principles to map the informal CIG metamodel to a precise specification. Due to OWL's rather verbose syntax, and OWL 2 Description Logics (DL) being essentially a serialisation of the $\mathcal{ALCQ}$ DL language, I present the CIG metamodel's main axioms in DL notation to demonstrate its feasibility, with the usual semantics as defined in [284]. In the DL notation, $\equiv$ means 'equivalent', $\sqcap$ 'and', $\exists$ 'at least one', $= n$ 'only n, where n is an integer', and $\forall$ 'for all'. All classes are disjoint (axioms not included). The full specification in OWL is presented in Appendix C. The DL statements use the class and association names specified in Fig. 5.3. Each axiom in the formalisation listed below is illustrated with examples from the Malawi HIV guidelines.

$$\begin{aligned}\mathsf{Guideline} \equiv &(\exists \mathsf{hasRecommendation.Recommendation} \sqcap \\ &\forall \mathsf{hasRecommendation.Recommendation}) \sqcap \\ &(\exists \mathsf{hasCondition.Condition} \sqcap \forall \mathsf{hasCondition.Condition}) \sqcap \\ &(\exists \mathsf{hasAction.Action} \sqcap \forall \mathsf{hasAction.Action})\end{aligned} \tag{5.1}$$

$$\begin{aligned}\mathsf{Recommendation} \equiv &(\exists \mathsf{referencesCondition.Condition} \sqcap \\ &\forall \mathsf{referencesCondition.Condition}) \sqcap \\ &(\exists \mathsf{referencesAction.Action} \sqcap \forall \mathsf{referencesAction.Action})\end{aligned} \tag{5.2}$$

$$\begin{aligned}\mathsf{Condition} \equiv &(= 1\ \ \mathsf{hasDecisionVariable.DecisionVariable} \sqcap \\ &\forall \mathsf{hasDecisionVariable.DecisionVariable}) \sqcap \\ &(= 1\ \ \mathsf{hasRelator.Relator} \sqcap \forall \mathsf{hasRelator.Relator}) \sqcap \\ &(= 1\ \mathsf{hasVariableValue.VariableValue} \sqcap \\ &\forall \mathsf{hasVariableValue.VariableValue})\end{aligned} \tag{5.3}$$

$$\mathsf{ConditionWithUnit} \equiv \mathsf{Condition} \sqcap (= 1\ \mathsf{hasUnit.Unit}) \tag{5.4}$$

$$\begin{aligned}\mathsf{Action} \equiv &(= 1\ \mathsf{hasActionVerb.ActionVerb} \sqcap \forall \mathsf{hasActionVerb.ActionVerb}) \sqcap \\ &(= 1\ \mathsf{hasActionVerbComplement.ActionVerbComplement} \sqcap \\ &\forall \mathsf{hasActionVerbComplement.ActionVerbComplement})\end{aligned} \tag{5.5}$$

First, a clinical practice guideline denoted by Guideline is defined in Eq. 5.1, i.e., a Guideline has *at least one* guideline recommendation denoted by Recommendation, and only guideline recommendations. Guideline has as instance, e.g., *"Definition of ART eligibility"* in the *"2014 Malawi Integrated Guidelines for Providing HIV Services"*. An example of a Recommendation is one for managing an *"Infant under 12 months: Confirmed HIV infection (DNA–PCR needed)"*. A Recommendation consist of at least one condition, denoted by Condition and at least one recommended action, denoted by Action to be executed on a patient when the all the conditions in the Guideline are satisfied (Eq. 5.2). For instance, *"HIV test result is positive"* is a Condition and *"Prescribe regimen 4"* is an example of an Action. Delving into conditions, there are: decision variables, denoted by DecisionVariable; relators, denoted by Relator; and decision variable values, denoted by VariableValue, satisfying the constraints as in Eq. 5.3. For instance, take the Condition *"HIV test result is positive"*, which has as DecisionVariable *"HIV test result"*, Relator *"is"*, and a VariableValue *"positive"*. One can also have a unit, denoted by Unit, associated with conditions that have units denoted by ConditionWithUnit, as in Eq. 5.4; e.g., in the Condition *"Age more than five months"*, *"months"* is a Unit. Finally, a recommended action, denoted by Action, can be defined as in Eq. 5.5, availing of an action verb denoted by ActionVerb and an action verb complement, denoted by ActionVerbComplement. For instance, *"Prescribe regimen 4"* is an Action, in which the ActionVerb is *"Prescribe"* and the ActionVerbComplement is then *"regimen 4"*.

5.6 The evolving CIG language model

In order to create a precise specification of a sufficient abstract grammar that can model an evolving CIG, I created a formal language model as is with common practice [153, 181, 186–188] using concepts, relationships and constraints from the previously specified evolving CIG conceptual model. The sections that follow detail the denotational semantics, abstract syntax and evaluation semantics of the evolving CIG modelling grammar.

5.6.1 Denotational semantics for the evolving CIG modelling language

I started by defining semantic functions that map each of the axioms of the evolving CIG conceptual model in Eqs 5.1–5.5 into meanings that constructors of an abstract grammar should denote. I adopted Reynold's [186] semantic equation theories and applied them to specify denotational semantics that suit our evolving CIG modelling case. The semantic equations were required so that the set of symbols or expressions that are required by each constructor in the evolving CIG modelling language grammar could be identified.

When a semantic equation rule is displayed for a linguistic construction, the kind of equation as well as the type of construction is going to be indicated. For instance, DR SEM EQ abbreviates *direct semantic equation*. Italic letters are used for metavariables and a sans serif font is used for object variables. Metavariables, in the metalanguage, indicate what type of entity they range over through the common mathematical convention. For instance, e ranges over expressions; v and w range over object variables; k, m and n over integers; η over environments; and p will range over phrases of several types. If D_∞ is a domain that contains more than one element, the value of an expression depends on the values of its free variables. The meaning of an expression is a continuous function from $D_\infty^{\langle var \rangle}$ to D_∞:

$$[\![-]\!] \in \langle exp \rangle \rightarrow (D_\infty^{\langle var \rangle} \rightarrow D_\infty) \tag{5.6}$$

All members of $D_\infty^{\langle var \rangle}$ are environments. The function ϕ or ψ is used to convert a value into a function, or vice versa, where necessary. The expression

$$\langle x'_0, ..., x'_{n-1} \rangle \quad \text{for} \quad [0 : x'_0|...|n-1 : x'_{n-1}], \tag{5.7}$$

denotes the function with domain 0 **to** $n-1$, called an *n-tuple*, that maps each i into x'_i. [186]

I started by defining the canonical forms that correspond to each semantic element of the evolving CIG conceptual model. I define Eq. 5.8 for DecisionVariable, Eq. 5.9 for Relator, Eq. 5.10 for VariableValue, Eq. 5.11 for Unit, Eq. 5.12 for ActionVerb and Eq. 5.13 for ActionVerbComplement. These semantic elements can be constructed by their respective functions over a variable whose domain is an infinite set of variables of unspecified representations. Thereafter, I defined Eq. 5.14 for Condition as a function over three expressions that depict DecisionVariable, Relator and VariableValue elements; and Eq. 5.15 for ConditionWithUnit as a function over four expressions that depict DecisionVariable, Relator, VariableValue and Unit elements. Furthermore, I defined Eq. 5.17 for Recommendation as a function over two sets of n-tuple of expressions where one set is for Condition and the other set is for Action elements. I finally defined Eq. 5.18 for Guideline as function over three sets of n-tuple of expressions that depict the first set for Condition elements, the second set for Action elements, and the third set for Recommendation elements.

Table 5.2 shows the mapping between the axioms of the conceptual model in Eqs 5.1–5.5 to their corresponding denotational semantics in equations 5.8–5.18. The semantic functions are defined by their associated semantic equations as follows:

TABLE 5.2: Conceptual model to semantic equation mapping

Concept name	**Conceptual Model Eq.**	**Semantic Eq.**
Guideline	Eq. 5.1	Eq. 5.18
Recommendation	Eq. 5.2	Eq. 5.17
Condition	Eq. 5.3	Eq. 5.14
ConditionWithUnit	Eq. 5.4	Eq. 5.15
Action	Eq. 5.5	Eq. 5.16

DR SEM EQ: DecisionVariable

$$[\![v]\!]\eta = \eta v, \tag{5.8}$$

DR SEM EQ: Relator

$$[\![v]\!]\eta = \eta v, \tag{5.9}$$

DR SEM EQ: VariableValue

$$[\![v]\!]\eta = \eta v, \tag{5.10}$$

DR SEM EQ: Unit

$$[\![v]\!]\eta = \eta v, \tag{5.11}$$

DR SEM EQ: ActionVerb

$$[\![v]\!]\eta = \eta v, \tag{5.12}$$

DR SEM EQ: ActionVerbComplement

$$[\![v]\!]\eta = \eta v, \tag{5.13}$$

DR SEM EQ: Condition

$$[\![e_0 e_1 e_2]\!]\eta = \phi([\![e_0]\!]\eta)([\![e_1]\!]\eta)([\![e_2]\!]\eta), \tag{5.14}$$

DR SEM EQ: ConditionWithUnit

$$[\![e_0 e_1 e_2 e_3]\!]\eta = \phi([\![e_0]\!]\eta)([\![e_1]\!]\eta)([\![e_2]\!]\eta)([\![e_3]\!]\eta), \tag{5.15}$$

DR SEM EQ: Action

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)([\![e_1]\!]\eta), \tag{5.16}$$

DR SEM EQ: Recommendation

$$[\![\langle e_0, ..., e_{m-1}\rangle\langle e'_0, ..., e'_{n-1}\rangle]\!]\eta = \phi([\![\phi([\![e_0]\!]\eta)...([\![e_{m-1}]\!]\eta)]\!]\eta)([\![\phi([\![e'_0]\!]\eta)...([\![e'_{n-1}]\!]\eta)]\!]\eta) \quad \textit{where } m, n > 0, \tag{5.17}$$

DR SEM EQ: Guideline

$$\begin{aligned}[\![\langle e_0, ..., e_{k-1}\rangle\langle e'_0, ..., e'_{m-1}\rangle\langle e''_0, ..., e''_{n-1}\rangle]\!]\eta &= \phi([\![\phi([\![e_0]\!]\eta)...([\![e_{k-1}]\!]\eta)]\!]\eta) \\ &([\![\phi([\![e'_0]\!]\eta)...([\![e'_{m-1}]\!]\eta)]\!]\eta)([\![\phi([\![e''_0]\!]\eta)...([\![e''_{n-1}]\!]\eta)]\!]\eta) \\ &\text{where } k, m, n > 0.\end{aligned} \tag{5.18}$$

5.6.2 Abstract syntax of the CIG metamodel

The abstract syntax of the CIG modelling language was realised by mapping each of the semantic equations presented in the previous section into constructors of an abstract grammar. Table 5.3 shows the mapping of the semantic equations to their corresponding constructors in the abstract grammar of the CIG DSL. I used Reynold's [186] abstract grammar specification theories to map the previously specified semantic equations into a sufficient abstract grammar for the evolving CIG modelling language.

Table 5.3: Semantic equations to abstract grammar mapping

Concept name	**Semantic Eq.**	**Abstract grammar Eq.**
DecisionVariable	Eq. 5.8	Eq. 5.19
Relator	Eq. 5.9	Eq. 5.20
VariableValue	Eq. 5.10	Eq. 5.21
Unit	Eq. 5.11	Eq. 5.22
Condition	Eq. 5.14	Eq. 5.23
ConditionWithUnit	Eq. 5.15	Eq. 5.24
ActionVerb	Eq. 5.12	Eq. 5.25
ActionVerbComplement	Eq. 5.13	Eq. 5.26
Action	Eq. 5.16	Eq. 5.27
Recommendation	Eq. 5.17	Eq. 5.28
Guideline	Eq. 5.18	Eq. 5.29

The phrases of a formal language are abstract entities such that specifying the semantics of the language can be achieved by defining semantic functions whose domains are sets of abstract phrases. Although phrases may be conceptually abstract, there is a need for their notation. $L ::= \rho_0|...|\rho_{k-1}$ abbreviates a set $\{L ::= \rho_0, ..., L ::= \rho_{k-1}\}$ of productions with the same left side. Each unabbreviated production has the form $L ::= s_0R_0...R_{n-1}s_n$, where $n \geq 0$, L and the R_i are nonterminals, and the s_i are strings of terminal symbols. Certain predefined nonterminals such as $\langle\mathsf{var}\rangle$ do not occur on the left side of productions. $\langle\mathsf{var}\rangle$ is a predefined nonterminal denoting a countably infinite set of variables with unspecified representations. Productions with the same left side always have distinct patterns of terminal symbols such that if $L ::= s_0R_0...R_{n-1}s_n$ and $L ::= s'_0R'_0...R'_{n'-1}s'_{n'}$, are distinct productions with the same left side, then either $n \neq n'$ or, for some $i \in 0$ to $n, s_i \neq s'_i$. For each nonterminal of the grammar, there must be a set of abstract phrases, called a *carrier*. There must be a function called a *constructor* for each production of the abstract grammar, such that, a production of the form $L ::= s_0R_0...R_{n-1}s_n$ gives rise to a constructor $c \in R_0 \times ... \times R_{n-1} \rightarrow L$. The operator $\rightarrow$ is right associative and has a lower precedence than $\times$. The carriers and constructors must satisfy the following: *i*) each of the constructors should be injective, *ii*) different constructors into the same carrier must have disjoint ranges, *iii*) each member of every carrier not predefined must be constructible using a finite number of applications of the constructors. [186]

Let the carriers of the productions for the abstract grammar of our CIG modelling language be among the following nonterminals: $\langle\mathsf{var}\rangle$, $\langle\mathsf{Unit}\rangle$, $\langle\mathsf{Relator}\rangle$, $\langle\mathsf{VariableValue}\rangle$, $\langle\mathsf{DecisionVariable}\rangle$, $\langle\mathsf{Condition}\rangle$, $\langle\mathsf{ActionVerb}\rangle$, $\langle\mathsf{ActionVerbComplement}\rangle$, $\langle\mathsf{Action}\rangle$, $\langle\mathsf{Recommendation}\rangle$, and $\langle\mathsf{Guideline}\rangle$.

The constructors for the abstract grammar, depicted as $c_{\langle element\rangle}$ where $\langle element\rangle$ is a semantic element of a CIG mapped from the semantic equations presented in Section 5.5.2, be as follows:

$$
\begin{aligned}
c_{DecisionVariable} \in& \langle \mathsf{var} \rangle \rightarrow \langle \mathsf{DecisionVariable} \rangle & (5.19)\\
c_{Relator} \in& \langle \mathsf{var} \rangle \rightarrow \langle \mathsf{Relator} \rangle & (5.20)\\
c_{VariableValue} \in& \langle \mathsf{var} \rangle \rightarrow \langle \mathsf{VariableValue} \rangle & (5.21)\\
c_{Unit} \in& \langle \mathsf{var} \rangle \rightarrow \langle \mathsf{Unit} \rangle & (5.22)\\
c_{Condition} \in& \langle \mathsf{DecisionVariable} \rangle \times \langle \mathsf{Relator} \rangle \times \langle \mathsf{VariableValue} \rangle &\\
& \rightarrow \langle \mathsf{Condition} \rangle & (5.23)\\
c_{ConditionWithUnit} \in& \langle \mathsf{DecisionVariable} \rangle \times \langle \mathsf{Relator} \rangle \times \langle \mathsf{VariableValue} \rangle \times &\\
& \langle \mathsf{Unit} \rangle \rightarrow \langle \mathsf{Condition} \rangle & (5.24)\\
c_{ActionVerb} \in& \langle \mathsf{var} \rangle \rightarrow \langle \mathsf{ActionVerb} \rangle & (5.25)\\
c_{ActionVerbComplement} \in& \langle \mathsf{var} \rangle \rightarrow \langle \mathsf{ActionVerbComplement} \rangle & (5.26)\\
c_{Action} \in& \langle \mathsf{ActionVerb} \rangle \times \langle \mathsf{ActionVerbComplement} \rangle &\\
& \rightarrow \langle \mathsf{Action} \rangle & (5.27)\\
c_{Recommendation} \in& \{\langle \mathsf{Condition} \rangle_0, ..., \langle \mathsf{Condition} \rangle_{m-1}\} &\\
& \times \{\langle \mathsf{Action} \rangle_0, ..., \langle \mathsf{Action} \rangle_{n-1}\} &\\
& \rightarrow \langle \mathsf{Recommendation} \rangle &\\
& \text{ where } m, n > 0 & (5.28)
\end{aligned}
$$

$$
\begin{aligned}
c_{Guideline} \in& \{\langle \mathsf{Recommendation} \rangle_0, ..., \langle \mathsf{Recommendation} \rangle_{k-1}\} &\\
& \times \{\langle \mathsf{Condition} \rangle_0, ..., \langle \mathsf{Condition} \rangle_{m-1}\} &\\
& \times \{\langle \mathsf{Action} \rangle_0, ..., \langle \mathsf{Action} \rangle_{n-1}\} &\\
& \rightarrow \langle \mathsf{Guideline} \rangle &\\
& \text{ where } k, m, n > 0 & (5.29)
\end{aligned}
$$

I start by mapping Eq. 5.8 to Eq. 5.19 for DecisionVariable, Eq. 5.8 to Eq. 5.20 for Relator, Eq. 5.8 to Eq. 5.21 for VariableValue, Eq. 5.8 to Eq. 5.22 for Unit, Eq. 5.8 to Eq. 5.25 for ActionVerb and Eq. 5.8 to Eq. 5.26 for ActionVerbComplement such that each of the resulting constructors create its respective CIG semantic element from a variable whose domain is an infinite set of variables of unspecified sequence. I further map Eq. 5.14 to Eq. 5.23 that constructs Condition as a function of DecisionVariable, Relator and VariableValue. Eq. 5.15 to Eq. 5.24 to define a constructor that constructs ConditionWithUnit as a function of DecisionVariable, Relator, VariableValue and Unit. Thereafter, I mapped Eq. 5.16 to Eq. 5.27 to define a constructor that constructs Action as a function of ActionVerb and ActionVerbComplement. I further map Eq. 5.17 to Eq. 5.28 that construct

Recommendation as function of a Condition tuple and an Action tuple. I finally map Eq. 5.18 to Eq. 5.29 to define a constructor for Guideline as a function of Condition, Action and Recommendation tuples.

5.6.3 Evaluation semantics

Evaluation semantics, which are also known as natural semantics, can describe the behaviour of software languages such as DSLs and programming languages [153, 186, 285]. In order to define evaluation semantics of the evolving CIG modelling language grammar, I create a specification using evaluation contexts as is with common practice [187, 285, 286]. Let T be a semantic element of an evolving CIG, and a list of the semantic elements be $List(T)$(e.g. $List(\text{Recommendation})$ referring to a list of Recommendation elements). I define three evaluation contexts that are of interest to CIG evaluation.

I define the first evaluation context E_c for evaluating a Condition as a function on a patient record P and a Condition so that the function returns a boolean true or false depending on whether the condition is valid on the patient or not. Let *bool* be a boolean $TRUE$ or $FALSE$. The evaluation context for evaluating a particular Condition is defined as:

$E_c \in P \times \text{ Condition } \rightarrow bool$

I define the second CIG evaluation context E_g as a function on a patient record P and a particular Guideline so that the function returns a list of applicable Action elements from the Guideline. The evaluation context for evaluating a Guideline is defined as:

$E_g \in P \times \text{ Guideline } \rightarrow List(\text{Action})$

The evolving CIG evaluation context E_g allows an actor within a clinical decision support system to evaluate an entire Guideline with a particular patient record and obtain a list of applicable Action elements.

I define the third evolving CIG evaluation context E_r as a function of a patient record P and a particular Recommendation so that the function returns a list of applicable Action elements from the Recommendation. The evaluation context for evaluating a Recomendation is defined as:

$E_r \in P \times \text{ Recommendation } \rightarrow List(\text{Action})$

The CIG evaluation context E_r allows an actor within a clinical decision support system to evaluate a particular Recommendation with a particular patient record and obtain a list of applicable Action elements.

5.7 A concrete syntax and tool implementation for modelling evolving CIGs

In this section, I use a context-free grammar to define terminal and production rules of the evolving CIG modelling language grammar. I use Xtext grammar language, an Extended Backus-Naur Form-like notation, designed for the description of concrete syntaxes for textual languages. Xtext grammar language is extended by object-oriented concepts and features that are used for the automatic generation of metamodels, parsers and related editors [287]. A context-free grammar was adopted because of the following: *i*) is an established method that allows a precise syntactic specification of a software language; *ii*) supports the introduction of new and the maintenance of existing features as a software language evolves; *iii*) and supports the automatic construction of parsers and language editors [153, 181, 288]. I use Bettini's [207] methods to map the abstract grammar of the CIG modelling language that is presented in Section 5.6.2 into a concrete syntax that I call *FCIG*. I used Bettini's methods because they provide quick and easy methods, that are based on best practices, for mapping formal constructs of a software language, into a working implementation using the Xtext framework [207].

Xtext can infer *Ecore* models from a grammar. A token is an atomic symbol that consists of one or more characters that is matched by a particular terminal rule or keyword. Terminal rules in the grammar language, also referred to as lexer rules or token rules, return string (type *ecore::Estring*) by default although it is possible to return any type that is an instance of *ecore::EDataType*. There are various kinds of token rule expressions. Each token rule expression can have one of the four cardinalities as follows: *i*) exactly one, which is the default, no operator; *ii*) one or none, operator ?; *iii*) zero or more, operator *; *iv*) one or more, operator +. Keywords are terminal rule literals that can contain arbitrary characters and can be of any length. Character ranges can be declared using the *'..'* operator. The wildcard operator *'.'* can be used to allow any character in particular position of a character sequence. The until operator *'–⟩'* allows one to state that everything should be consumed until a particular token occurs. The negated operator *'!'* can be used to invert any tokens that immediately succeed it. The grammar language allows rules to refer to other rules by writing the name of the rule to be called. Rule calls in terminal rules can only be make calls to other terminal rules. The *'|'* operator allows one to state multiple different alternatives. [287]

When a parser reads a sequence of terminals and walks through a parser rule, a parse tree that consists of non-terminal and terminal tokens is produced. Parser rules are also called production rules. The character ranges, wildcard, until token and negation operators are only available for terminal rules. Furthermore, the rest of the concepts and syntactical constructs available for terminal rules are also available for production rules. In parser rules, assignments are used to assign parsed information to a feature of the object that is in current scope. There are three different types of assignment operators: *i*) The '=' operator, for straight forward assignment *ii*) The '+=' operator, which adds the value on the right to a multi-valued feature *iii*) The '?=' operator, which sets a feature of type *ecore::EBoolean* to true if the right hand side was consumed independently from the concrete value of the right hand side. The Xtext grammar language allows one to declare crosslinks in the grammar by placing a type between square brackets. [287]

$$\begin{aligned} \textit{Guideline:}\quad & \textit{st+=(Condition} \mid \textit{Action)*} \\ & \textit{st+=(Recommendation)+;} \end{aligned} \tag{5.30}$$

$$\begin{aligned} \textit{Recommendation:}\quad & \textit{'Recommendation' name=ID ':'} \\ & \textit{'Conditions'} \\ & \textit{conditions+=[Condition](',' conditions+=[Condition])*} \\ & \textit{'Actions'} \\ & \textit{actions+=[Action](',' actions+=[Action])*;} \end{aligned} \tag{5.31}$$

```
                 Condition: ’Condition’ name=ID ’:’
                            decisionVariable=DecisionVariable
                            relator=Relator
                            variableValue=VariableValue
                            (unit=Unit)?;                                        (5.32)
                    Action: ’Action’ name=ID ’:’
                            actionVerb=ActionVerb
                            actionVerbComplement=ActionVerbComplement;           (5.33)
          DecisionVariable: value = STRING |
                            value = ID;                                          (5.34)
                   Relator: value= ’is’ | value= ’=’ |
                            value = ’〉’ | value = ’〉=’ |
                            value = ’〈’ | value = ’〈=’;                          (5.35)
             VariableValue: value = NUMBER | value = STRING |
                            value = ’true’ | value = ’false’ |
                            value = ID;                                          (5.36)
                      Unit: value = STRING | value = ID;                         (5.37)
                ActionVerb: value = ID | value = STRING;                         (5.38)
      ActionVerbComplement: value = ID | value = STRING;                         (5.39)
               terminal ID: ’^’?(’a’..’z’|’A’..’Z’|’_’)
                            (’a’..’z’|’A’..’Z’|’_’|’0’..’9’)*;                   (5.40)
           terminal STRING: ’”’ ( ’\\’ . /* ’b’|’t’|’n’|’f’|’r’|’u’|’”’|”’”|’\\’ */ | !(’
                            \\’|’”’) )* ’”’ |
                            ”’” ( ’\\’ . /* ’b’|’t’|’n’|’f’|’r’|’u’|’”’|”’”|’\\’ */ | !(’
                            \\’|”’”) )* ”’”;                                     (5.41)
       terminal ML_COMMENT: ’/*’ -〉 ’*/’;                                        (5.42)
       terminal SL_COMMENT: ’//’ !(’\n’|’\r’)* (’\r’? ’\n’)?;                    (5.43)
               terminal WS: (’ ’|’\t’|’\r’|’\n’)+;                               (5.44)
           terminal NUMBER returns ecore::EString:
                            (’-’)?(’0’..’9’)* (’.’ (’0’..’9’)+)?;                (5.45)
```

I mapped the concrete syntax grammar of the CIG modelling language from the abstract syntax grammar specified in Section 5.6.2. I further use the Xtext framework with the Eclipse Modelling Framework (EMF) [199] to implement the concrete syntax grammar and build appropriate language support tools for the CIG modelling language. The terminal and production rules of a concrete syntax grammar, that I named *FCIG*, are presented in Eqs. 5.30–5.45. I reused some of the terminal rules from the default set of terminal rules of the Xtext framework. I specifically reused the terminal rules for identifiers, strings, single line comments, multiple line comments and white spaces specified in Eqs. 5.40–5.44. To complete the specification of terminal rules for the grammar, I defined a new terminal rule for both positive and negative decimal numbers in Eq. 5.45. I mapped the rest of the production rules in the concrete syntax grammar specification from their corresponding constructors in the abstract grammar presented in Section 5.6.2. Table 5.4 shows the mapping between the abstract grammar and concrete syntax specification.

Table 5.4: Abstract grammar to concrete syntax mapping

.

Constructor	**Abstract grammar Eq.**	**Concrete syntax Eq.**
Guideline	Eq. 5.29	Eq. 5.30
Recommendation	Eq. 5.28	Eq. 5.31
Condition, ConditionWithUnit	Eqs. 5.23-5.24	Eq. 5.32
DecisionVariable	Eq. 5.19	Eq. 5.34
Relator	Eq. 5.28	Eq. 5.31
VariableValue	Eq. 5.21	Eq. 5.36
Unit	Eq. 5.22	Eq. 5.37
Action	Eq. 5.27	Eq. 5.33
ActionVerb	Eq. 5.25	Eq. 5.38
ActionVerbComplement	Eq. 5.26	Eq. 5.39

To start with, Eq. 5.29 is mapped to Eq. 5.30 which defines Guideline as the root element of the AST for the CIG modelling DSL. Eq. 5.30 further declares that Guideline is a collection of Condition, Action and Recommendation elements. Eq. 5.28 is mapped to Eq. 5.31 defining a Recommendation as an element that references one or more of existing Condition and Action elements. Eqs. 5.23-5.24 are mapped to Eq. 5.32 defining that a Condition element is made up of DecisionVariable, Relator, VariableValue and optional Unit elements. Eq. 5.32 further declares that the Condition construct begins with the *'Condition'* keyword that is followed by the name of the condition succeeded by semicolon and a space separated list of DecisionVariable, Relator, VariableValue and Unit elements. Eq. 5.27 is mapped to Eq. 5.33 declaring that an Action element is made up of ActionVerb and ActionVerbComplement elements. Eq. 5.33 further declares that the Action construct begins with the *'Action'* keyword followed by the name of the Action

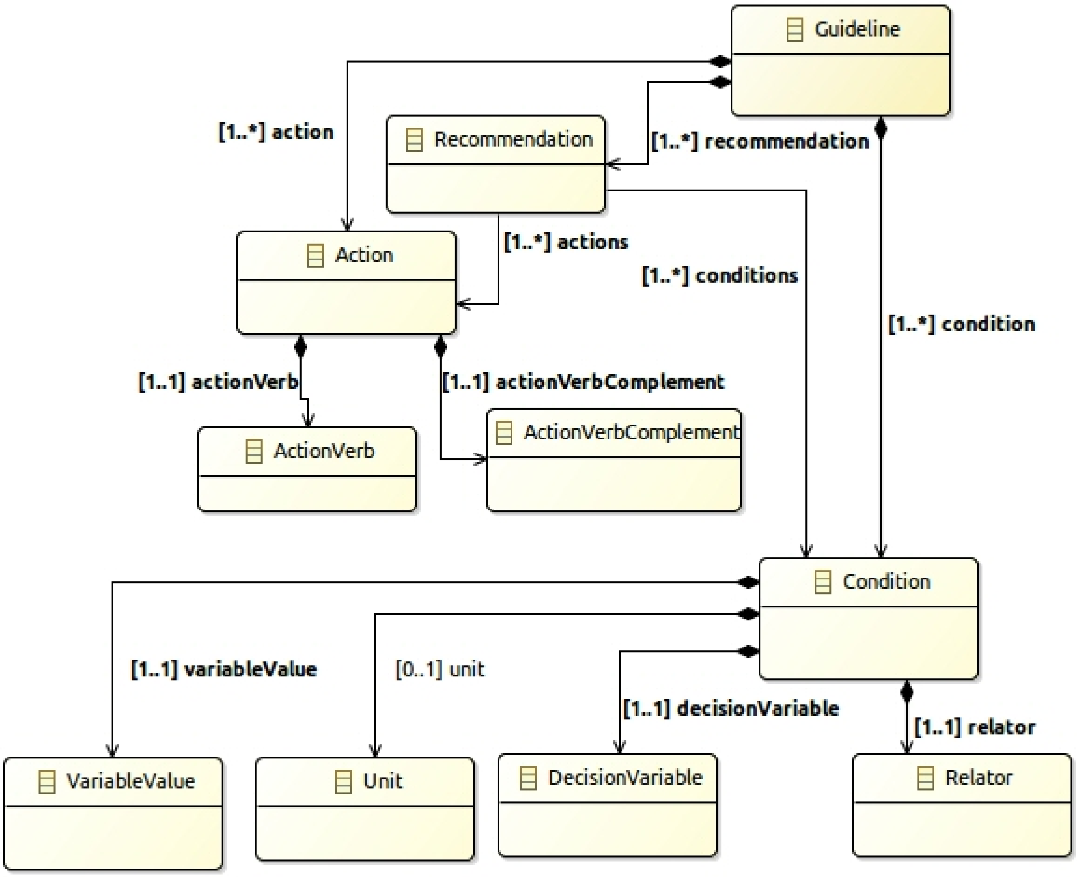


FIGURE 5.4: An EMF metamodel generated from the concrete syntax implementation in Xtext

succeeded by a semicolon and a space separated pair of ActionVerb and VerbComplement elements. Each of the constructors for the carriers in Eqs. 5.23-5.24 are mapped to their concrete syntax productions as follows: *i*) Eq. 5.19 is mapped to Eq. 5.34 for DecisionVariable, *ii*) Eq. 5.20 is mapped to Eq. 5.35 for Relator, *iii*) Eq. 5.21 is mapped to Eq. 5.36 for VariableValue, *iv*) Eq. 5.22 is mapped to Eq. 5.37 for Unit. And each of the constructors for the carriers in Eq. 5.27 are mapped to their concrete syntax productions as follows: *i*) Eq. 5.25 is mapped to Eq. 5.38 for ActionVerb, *ii*) Eq. 5.26 is mapped to Eq. 5.39 for ActionVerbComplement. The complete concrete syntax grammar in Xtext grammar language is presented in Appendix B. Figure 5.4 shows a class diagram of the EMF metamodel, at layer M2 of the CIG modelling architecture of Fig. 5.2, generated from the concrete syntax grammar implementation with the Xtext framework. It can be noted that the EMF metamodel of *FCIG* corresponds to the evolving CIG conceptual model in that all the concepts that are specified in the evolving CIG conceptual model can be traced by name apart from the concept ConditionWithUnit. But because a ConditionWithUnit is also a Condition, it can be traced as a Condition with an optional Unit element in the EMF metamodel.

I built a lexer, parser, abstract syntax tree (AST) model, smart editor plugin that includes full language support with features such as code completion, syntax highlighting, code folding, immediate feedback, incremental syntax checking by implementing the concrete syntax grammar using the Xtext and EMF frameworks.

5.8 Code generation for guideline-based clinical decision support

In this section, I describe the code generation approach for generating application-specific code for layer M0 of the CIG modelling architecture. The generated code conforms to CIG models at Layer M1 of the CIG modelling architecture. We can navigate the abstract syntax tree (AST) of a particular CIG model to generate code that is compatible with a specific clinical decision support systems application architecture. I use the following generalised specification to describe the fundamental components of a sufficient code generator for CIG models in the four-layer CIG modelling architecture.

For a Guideline in the AST of an evolving CIG model:

1. Create an empty sourcecode file that corresponds to the Guideline.
2. Retrieve all Recommendations from the Guideline
3. For each retrieved Recommendation, generate a function that evaluates applicability of the Recommendation and returns a collection of Actions in the target programming language as follows:
   (a) Create an empty collection of Actions
   (b) For each Condition of the Recommendation, evaluate its DecisionVariable, Relator, Value and optional Unit tuple whether it is true for the instance
   (c) When all Conditions are true:
      i. Retrieve all Actions from the Recommendation
      ii. For each Action, append its ActionVerb, VerbComplement and Recommendation triple to the collection of Actions
   (d) Return the collection of Actions
4. Generate a function that evaluates all Recommendations from the Guideline and returns a collection of all applicable Actions

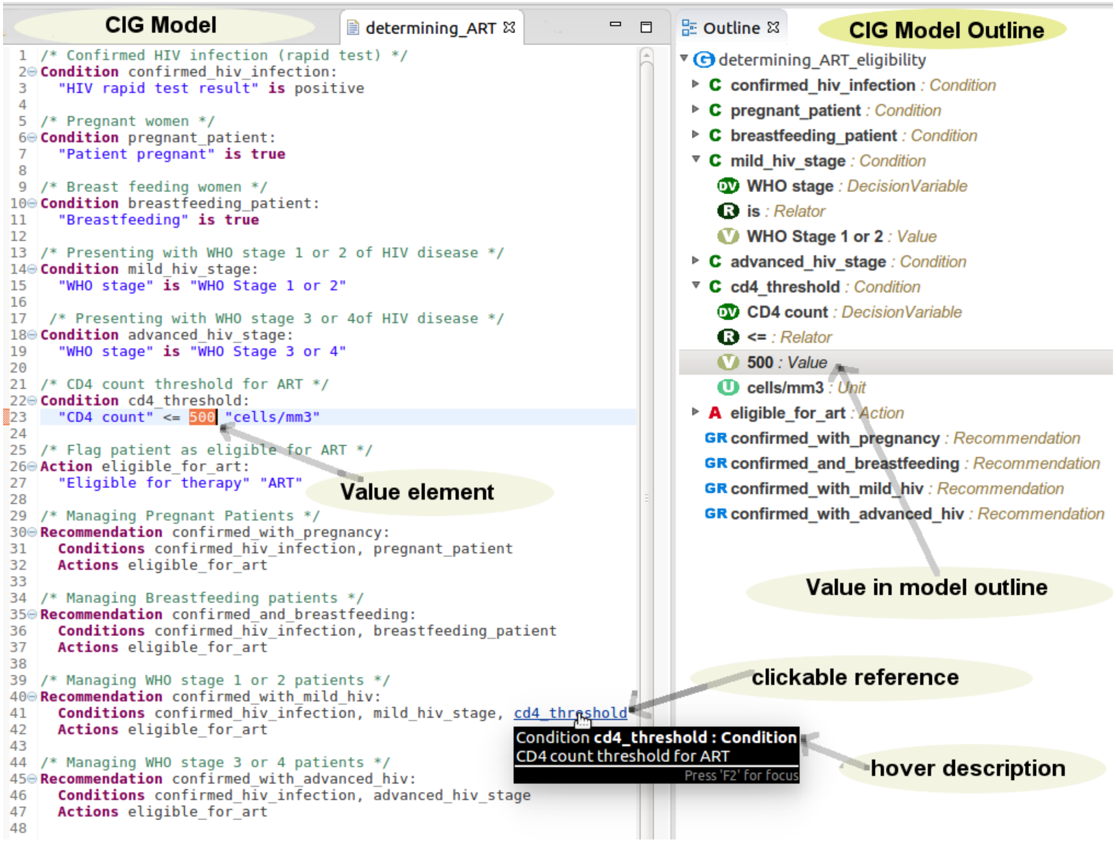
I demonstrated the feasibility of implementing such a code generator using the Xtext framework. I started by creating a prototype implementation of a guideline-based EMR system by extending the existing Malawi national ART system with a guideline recommendation evaluator capable of evaluating an evolving CIG based on the specification that were presented in Section 5.6.3. The sourcecode for the revised Malawi national ART system is on a github repository[2]. I further extended the Eclipse IDE plugin described in Section 5.7 with a code generator that generates evolving CIG libraries that are compatible with the patient reminders and alerts component of the Malawi national ART System.

5.9 Case Study

This section uses a case study to demonstrate the feasibility of managing a change using the CIG modelling architecture. We revisit the problem shown in Figure 5.1 (Section 5.2) and show that it now can be done in an semi-automated fashion with the DSL. In order to change the Value of the '*cd4_threshold*' Condition, the modeller selects the '*confirmed_with_mild_hiv*' Recommendation from the '*determining_ART_eligibility*' CIG model outline. Once input control is directed to the Recommendation definition, the '*cd4_threshold*' Condition is identified and verified through its reference and hover information. The modeller then clicks on the clickable reference (by holding the "Ctrl" button and clicking the link) to be taken to the '*cd4_threshold*' Condition definition. Once input control is directed to the Condition, the modeller can double click on the Value element of the Condition in the CIG model outline so input control is directed to the value specification for overwriting. Thereafter, the Value is overwritten with the desired value, in this case *500.* The CIG model can be saved once the modeller is satisfied with the update.

Fig. 5.5 shows an annotated screenshot of the resulting CIG model in the new CIG modelling environment. The Value element is clearly identifiable, accessible and manageable as a semantic element in the CIG model using an IDE. This process can be repeated for other changes up until the application of changes is completed. Once the changes are applied and the model is error-free, the code-generator automatically generates an updated CIG code library as specified in the code-generator. The generated code library was then bundled and compiled with the Malawi national ART system. Figure 5.6 shows the alerts and reminders window on a patient dashboard for an HIV positive patient whose CD4 count is 400 before and after the change was applied to the generated code library.

[2] https://github.com/yamiko/bart2_fcig

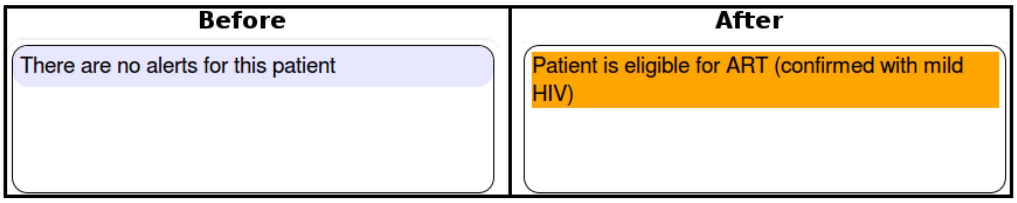


```
/* Confirmed HIV infection (rapid test) */
Condition confirmed_hiv_infection:
  "HIV rapid test result" is positive

/* Pregnant women */
Condition pregnant_patient:
  "Patient pregnant" is true

/* Breast feeding women */
Condition breastfeeding_patient:
  "Breastfeeding" is true

/* Presenting with WHO stage 1 or 2 of HIV disease */
Condition mild_hiv_stage:
  "WHO stage" is "WHO Stage 1 or 2"

 /* Presenting with WHO stage 3 or 4of HIV disease */
Condition advanced_hiv_stage:
  "WHO stage" is "WHO Stage 3 or 4"

/* CD4 count threshold for ART */
Condition cd4_threshold:
  "CD4 count" <= 500 "cells/mm3"

/* Flag patient as eligible for ART */
Action eligible_for_art:
  "Eligible for therapy" "ART"

/* Managing Pregnant Patients */
Recommendation confirmed_with_pregnancy:
  Conditions confirmed_hiv_infection, pregnant_patient
  Actions eligible_for_art

/* Managing Breastfeeding patients */
Recommendation confirmed_and_breastfeeding:
  Conditions confirmed_hiv_infection, breastfeeding_patient
  Actions eligible_for_art

/* Managing WHO stage 1 or 2 patients */
Recommendation confirmed_with_mild_hiv:
  Conditions confirmed_hiv_infection, mild_hiv_stage, cd4_threshold
  Actions eligible_for_art

/* Managing WHO stage 3 or 4 patients */
Recommendation confirmed_with_advanced_hiv:
  Conditions confirmed_hiv_infection, advanced_hiv_stage
  Actions eligible_for_art

```

Condition **cd4_threshold : Condition**
CD4 count threshold for ART
Press 'F2' for focus

CIG Model
determining_ART
Outline
CIG Model Outline
determining_ART_eligibility
C confirmed_hiv_infection : Condition
C pregnant_patient : Condition
C breastfeeding_patient : Condition
C mild_hiv_stage : Condition
DV WHO stage : DecisionVariable
R is : Relator
V WHO Stage 1 or 2 : Value
C advanced_hiv_stage : Condition
C cd4_threshold : Condition
DV CD4 count : DecisionVariable
R <= : Relator
V 500 : Value
U cells/mm3 : Unit
A eligible_for_art : Action
GR confirmed_with_pregnancy : Recommendation
GR confirmed_and_breastfeeding : Recommendation
GR confirmed_with_mild_hiv : Recommendation
GR confirmed_with_advanced_hiv : Recommendation
Value element
Value in model outline
clickable reference
hover description

FIGURE 5.5: An annotated screenshot of a CIG modelling environment that shows a CIG model on the left and its model outline on the right



There are no alerts for this patient

Patient is eligible for ART (confirmed with mild HIV)

FIGURE 5.6: Patient alerts window before and after a CPG change

5.10 Empirical results

In this section, I present results from the two evaluation studies. Recall the evaluation criteria used to address the second research question **RQ2**, through subquestions **RQ2-S1** and **RQ2-S2**, that was discussed in Section 5.3.4.

In order to address research question **RQ2-S1**, I recorded the number of guideline recommendations from the guideline strata that were adequately encoded using *FCIG*, in order to measure task completion rate. The hypothesis that guided my analyses for answering research question **RQ2-S1** were formulated as follows:

H$_0$: *FCIG* can not be used to complete CPG encoding tasks.

H_1: *FCIG* can be used to complete CPG encoding tasks adequately.

In order to address research question **RQ2-S2**, I recorded the total number of CPG changes that were directly supported with smart-editing features, in order to measure task completion rate. The hypothesis that guided my analyses for answering research question **RQ2-S2** were formulated as follows:

H_0: *FCIG* can not provide smart-editing support for CPG maintenance tasks.

H_1: *FCIG* can provide smart-editing support for CPG maintenance tasks.

5.10.1 Results from study one: Assessing *FCIG*'s CPG representation adequacy

The stratified random sampling technique resulted in a selection of 46 guideline recommendations from a total of 124 guideline recommendations identified in the 2008 Malawi integrated guidelines for the clinical management of HIV and the 2008 Malawi community IMCI guideline. Table 5.5 shows the guideline recommendations stratified by guideline. A detailed list of the representative sample of guideline recommendations is presented in Appendix D. All 46 guideline recommendations were adequately encoded using *FCIG* and the Malawi medical concept dictionary. The results allow us to reject the null hypothesis, $\mathbf{H}_0$, on CPG representation adequacy. Furthermore, the results allow for the alternative hypothesis $\mathbf{H}_1$: *FCIG* can be used to complete CPG encoding tasks adequately, to be accepted. Hence, we can say that *FCIG* has adequate capacity to encode CPGs. The encoded guidelines are available at the following online repository[3].

5.10.2 Results from study two: Assessing *FCIG*'s ability to support CPG changes

The stratified random sampling technique resulted in a selection of 44 CPG change incidents from a total of 171 change incidents identified between the 2008 and 2011 versions of the Malawi integrated guidelines for the clinical management of HIV. Table 5.6 shows the CPG change incidents stratified by CPG change type. A detailed list of the representative sample of guideline changes stratified by CPG change type is presented in Appendix E. All 44 CPG changes were successfully applied to their respective guideline recommendations of 2008. All the CIG semantic elements that were affected by the changes had corresponding *FCIG* language constructs with full smart-editing

[3] https://github.com/yamiko/chapter_5_study_one

Table 5.5: Guideline recommendations stratified by guideline

.

Guideline document	**Guideline**	**Number of recommendations**	**Weighted strata**
Malawi HIV 2008	Determining priority for CD4-lymphocyte count testing	7	3
	Determining ART eligibility	20	7
	Managing suspected ART drug failure	4	2
	Implementing standardised ART reviews	3	1
	Managing first-line drug reactions	6	2
	Managing first-line regimen in children	3	1
	Prescribing CPT	20	7
	Adult doses of ART	17	6
	Providing ART in special situations	13	5
	Managing ART patients who develop TB	5	2
	Managing Kaposi's Sarcoma	2	1
	Managing drug toxicity	4	2
	Managing symptoms during clinic visit	1	1
Malawi Community IMCI 2008	Community IMCI	19	6
Total		**124**	**46**

support. These results allow us to reject the null hypothesis $\mathbf{H}_0$ and accept the alternative hypothesis $\mathbf{H}_1$: *FCIG* can provide smart-editing support for CPG maintenance tasks. Therefore, we can say that *FCIG* enables smart-editing support for CPG update tasks. The *FCIG* encoded guidelines both before and after the changes were applied are available at the following online repository [4].

5.11 Discussion

The novel CIG modelling environment as an Eclipse plugin is the first of its kind to include all fine-grained components of changing CPGs. The details behind this CIG modelling environment, i.e., the rigorously defined evolving CIG conceptual model, denotational semantics, abstract syntax and concrete syntax within a CIG modelling architecture, are conveniently hidden for both types of prospective end users: CIG modellers

[4] https://github.com/yamiko/chapter_5_study_two

Table 5.6: CPG changes stratified by type of CPG change

Type of CPG change	**Change occurrences**	**Weighted strata**
Addition of a decision variable	3	1
Change of a decision variable value	13	3
Removal of a decision variable	2	1
Change of a decision variable	12	3
Addition of a recommended action	2	1
Removal of a recommended action	2	1
Change of an action verb complement	5	2
Change of a recommended action	2	1
Addition of a recommendation	55	13
Removal of a recommendation	75	18
Total	**171**	**44**

and EMR system implementers. The CIG modelling framework enables a CIG modeller to change only the fraction of an evolving CIG that has to be changed meanwhile enforcing the language implementations' conformity to the evolving CIG conceptual model, rather than having to rewrite most of the CPG and possibly introducing errors in the process. The CIG modelling framework further provides smart-editing features to support the CIG modeller in carrying out CPG change operations. The EMR system implementers now can spend less time on CIG maintenance and the tools that rely on it, for automated and semi-automated support for managing CPG changes can be provided.

Although *FCIG* has been created as a plugin for the Eclipse IDE, the Xtext language development framework is behind it. Practically, this means that one also can use the here presented concrete syntax to create other plugins that are targeted for other IDEs or web browser-based editors. This can avail smart-editing features that can support the modelling of evolving CIGs.

Observe that the CIG modelling language is comprised of language constructs that are based on existing concepts in the clinical domain and other CIG models, and thus are appropriate for modelling CIGs. The evolving CIG language model naturally fits within rule-based CIG DSLs that are well aligned with the modelling of public health CPGs which are a focus of this thesis.

The new evolving CIG conceptual model also naturally fits into the "if–then..." or trigger-action programming model. The trigger-action programming model is a workable solution that has been used in other domains such as smart homes because of its conceptual simplicity [289]. The new CPG metamodel with its explicit definition of fine-grained CPG elements may spurn the development of optional complex trigger-action constructs within CIG models. These complex trigger-action constructs might be explored for application by experienced users of the new CIG modelling language, which is an aspect of future work.

The use case demonstrated that hitherto manual operations can be carried out in the CIG modelling environment. While a single example may seem limited, it was a CPG from a common 'stock': many countries, including Malawi, adopt CPGs that are published by international organisations such as the World Health Organisation (WHO) and deploy these CPGs for routine clinical practice; e.g. countries in Latin America [290], Asia, and other countries in Africa [291]. Thus, it can be equally well used elsewhere.

Xtext generates ANother Tool for Language Recognition (ANTLR) [292] parsers for parsing context-free grammars [207]. ANTLR supports LL($*$) class of grammars that are based on left-to-right scanning and leftmost derivation algorithms [153]. Noting that LL parsers do not support left-recursion and are bounded by time and memory [153], the concrete syntax specification ensured that none of the language constructs for the new CIG modelling language were left-recursive. Furthermore, the lexing and parsing of the CIG models in the implementation of the IDE were found to be sufficient and efficient when editing CIG models.

5.12 Chapter summary

In this chapter, I started by describing a novel CIG modelling architecture whose foundations are centred on a formal evolving CIG conceptual model. I specified formal semantics of a language model that conforms with the evolving CIG conceptual model. Thereafter, I systematically mapped the DSL model into a CIG modelling language *FCIG*. I later evaluated *FCIG* within the novel CIG modelling architecture through scenario-based validation and an empirical study. *FCIG* and its modelling architecture were found to be adequate for modelling evolving CIGs. Furthermore, the evolving CIG modelling architecture was found to enable smart-editing features that can support CPG change operations. I further, demonstrated the feasibility of integrating the novel CIG modelling architecture with an EMR system for clinical decision support. In the next chapter, I describe the study that I carried out to evaluate the usability of the language constructs that make up the evolving CIG modelling language *FCIG*.

Chapter 6

FCIG grammar evaluation of perceived usability

6.1 Introduction

Computer-interpretable guideline (CIG) modelling languages form the interface between CIG modellers and the CIGs they require to be executed in computer-based clinical decision support systems. CPGs change frequently as the science and technology behind clinical practice improves. The CPG changes demand alterations in related CIGs as highlighted in a contextual inquiry that was carried out as part of this work in Section 2.4 of Chapter 2. These evolving CIGs require a CIG modelling language that can support computational management of such changes. Like other software languages, CIG modelling languages have rarely received usability or human factors evaluation which may have led to the deployment of inadequate languages [217].

In Chapter 5, I evaluated the grammar of a novel and compact CIG modelling language *FCIG* for its CPG representational adequacy and its support for enabling computational management of CPG changes as characterised in Chapter 4. In this chapter, I continued to evaluate *FCIG* by evaluating the usability of its language constructs. I began by assessing the perception of novice and experienced CIG modellers towards *FCIG*'s usability. I further highlighted key language characteristics and related evaluation tools that were perceived to contribute to *FCIG*'s usability.

6.2 Materials and methods

The aim of this study was to evaluate whether *FCIG*'s grammar is suitable and acceptable from a modeller's standpoint when modelling an evolving CIG. I evaluated user perceptions on the usability of *FCIG*'s language constructs.

6.2.1 Research questions

Recall the second research question **RQ2** in Section 3.1:

Can a model-driven engineering approach adequately support the modelling of an evolving CIG?

In this study, I continued to address the second research question **RQ2** from a CIG modeller's viewpoint. Recall, from the discussions in Chapter 2, that incompatible abstractions between language users and language designers can pose challenges to usability of modelling languages [213]. In addition, perceived usability is an important high-level construct of usability [231, 232]. In order to focus on this perspective, I set the following research subquestion:

RQ2-S3: *Are the language constructs of **FCIG** perceived as usable?*

6.2.2 Research design

I adopted a mixed methods approach for this evaluation. A mixed methods approach combines quantitative and qualitative techniques, methods, approaches, concepts or language into a single study [19]. A mixed methods approach can provide strengths from one method that can make up for the weaknesses in another method, thereby developing rich insights into phenomena of interest that can not be fully understood using a single method [293]. I separated the usability evaluation into two smaller studies, one with novice CIG modellers and the other with experienced CIG modellers as study participants. I collected quantitative through a standardised usability questionnaire that used Likert scales. I further collected qualitative data regarding grammar usability in order to understand the perception of CIG modellers when exposed to the language constructs that are provided in *FCIG*'s grammar.

6.2.3 Study setup

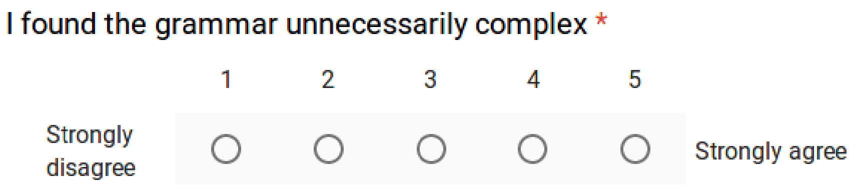
Study participants were presented with *FCIG*'s language constructs for evaluation. I gave all the study participants a short paper-based orientation of *FCIG*'s syntax. For each study participant, I started by explaining the purpose of the CIG modelling language. Thereafter, I gave a scripted description of the three main language constructs of *FCIG*. A copy of the paper-based orientation is available in Appendix F. Finally, the study participants evaluated the three main constructs of the grammar by responding to a usability questionnaire.

6.2.4 Study participants

Voluntary participation in the study was open to potential participants. Two categories of participants were recruited for the study through convenience sampling. The first category was that of novice CIG modellers with basic knowledge in computing but no experience in clinical decision support system design and deployment. I recruited the first category of study participants from the computer science postgraduate research laboratory at the University of Cape Town. The second category was that of individuals with prior experience in clinical decision support system design and deployment. I recruited the second category of study participants from my network of clinical decision support system designers and implementers. I planned to recruit at least five study participants in both categories of study participants.

6.2.5 Data collection methods

I used online questionnaires for data collection. In the questionnaires, I used the system usability scale (SUS) [248] that uses a five-point Likert scale as a standardised questionnaire to collect data about study participants' perceptions on the usability of *FCIG*'s grammar. Figure 6.1 shows one of the questions from the questionnaire on such a Likert scale. I chose SUS as a standardised questionnaire because SUS is a simple and widely



I found the grammar unnecessarily complex *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Figure 6.1: Sample question using a five-point Likert scale

used ten-item scale that gives a global view of subjective assessments of usability [248]. As discussed in Section 2.2.2 of Chapter 2, the SUS as a standardised usability questionnaire underwent considerable psychometric evaluations which showed some evidence of its validity, reliability and sensitivity [231]. In addition, the SUS does not require a user licence [231, 232]. I further included two additional open-ended questions to the online questionnaire in order to collect qualitative data that can give deeper insights where possible. Table 6.1 shows questions from number one to ten that were used for the SUS survey in addition to questions eleven and twelve for the qualitative survey.

Table 6.1: Questions for the survey

Question Number	**Text**
1	I think that I would like to use this grammar frequently
2	I found the grammar unnecessarily complex
3	I thought the grammar was easy to use
4	I think that I would need the support of a technical person to be able to use this grammar
5	I found the various functions in this language were well integrated
6	I thought there was too much inconsistency in this grammar
7	I would imagine that most people would learn to use this grammar very quickly
8	I found the grammar very cumbersome to use
9	I felt very confident using the grammar
10	I needed to learn a lot of things before I could get going with this grammar
11	Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?
12	Are there any features or keywords that are missing or need to be improved in the grammar?

6.2.6 Study design

I adopted a survey study for the evaluation. Survey studies are procedures in which investigators administer a survey to a sample or entire population to get a description of the attitudes, opinions, behaviours or characteristics of the population [21].

6.2.7 Study protocol

For this study, each participant went through the following procedure:

1. The purpose of the evaluation was explained to the participant

2. The participant completed an informed consent form

3. The participant was given a short paper-based orientation of the three main language constructs that make up *FCIG*'s grammar

4. The participant was given a link to complete a survey electronically

5. The participant completed the survey

After following the afore described procedure with all the participants, I carried out analyses and evaluations using the collected data.

6.2.8 Criteria to address the research subquestion RQ2-S3

In order to address research subquestion **RQ2-S3**, I calculated SUS scores from the conducted surveys on a continuous scale that measures modellers' perceptions on language usability. Since *FCIG*'s language constructs are based on terminology that is commonly used in the clinical knowledge engineering domain, I expected CIG modellers to perceive its usability with a more positive attitude. In order to guide my analyses I set two alternative hypotheses.

I formulated the following hypothesis on the levels of SUS scores:

H$_0$: CIG modellers do not perceive the language constructs of *FCIG* as usable.

H$_1$: CIG modellers perceive the language constructs of *FCIG* as usable.

As the SUS surveys were conducted with two categories of CIG modellers, I further formulated the following hypothesis on the SUS score differences:

H$_0$: There is no difference in SUS scores between the ratings from experienced CIG modellers and those from novice CIG modellers.

H$_1$: The SUS scores from experienced CIG modellers are higher than the SUS scores from novice modellers.

I used the *Shapiro-Wilk* test to test the samples of SUS scores from both experienced and novice CIG modellers for normality. In addition, the *Shapiro-Wilk* test helped in determining a relevant statistical test that helped in determining the statistical significance of observed differences between experienced and novice CIG modellers.

6.3 Results

In this section, I present the results from the studies that were carried out to evaluate *FCIG* for its usability.

6.3.1 Participants

There were six experienced CIG modellers and thirteen novice CIG modellers that participated in this study. Of the six experienced CIG modellers, one was recruited from South Africa whilst the rest were recruited from Malawi. The experienced modellers were university graduates with a computing background. In addition, all the experienced modellers worked for organisations that develop clinical decision support systems in low- and middle-income countries. All the novice CIG modellers were recruited from the Computer Science department at the University of Cape Town. Of the 13 CIG modellers, seven were students that were studying towards a doctoral degree whilst the rest were studying towards a master's degree. The sections that follow present the results from the analysis that was carried out on the data that was collected through the SUS and qualitative surveys completed by the CIG modellers.

6.3.2 Results from the SUS surveys

The sections that follow describe the results from the analysis of data that was collected from the usability study questionnaires. The raw data from the questionnaires is available in Appendix K.

6.3.2.1 SUS score levels

I began by testing the first null and first alternative hypotheses. Individual SUS scores were calculated from the surveys that were carried out with both experienced and novice CIG modellers. Table 6.4 shows the aggregate responses from the SUS survey with experienced CIG modellers. The sample of experienced CIG modellers had a mean of 89.17, a median of 91.25 and a mode of 90 for its SUS scores.

From the SUS scores that were calculated from the survey that was carried out with the novice CIG modellers, a mean of 79.23, a median of 80 and a mode of 80 were derived. Table 6.5 shows the aggregate scores on how novice CIG modellers rated *FCIG*'s grammar on the SUS questionnaire. Using the calculated SUS scores from both experienced and novice modellers, I created density plots that are presented in Figure 6.2 so I could visualise the results. Further to that, I also created a box plot for the resulting SUS scores that is presented in Figure 6.3. I created the density plots and box plot to carry out some exploratory analyses on the data that was collected through the questionnaire. The exploratory analyses revealed that the samples of SUS scores from both experienced and novice modellers might not have come from normally distributed populations. The

Table 6.2: Statistics for SUS scores on *FCIG* language constructs

CIG modeller level	**SUS score**		
	Mean	**Median**	**Mode**
Experienced	89.17	91.25	90
Novice	79.23	80	80

analyses further revealed that the sample of SUS scores from experienced CIG modellers were higher than those that were computed from novice CIG modellers.

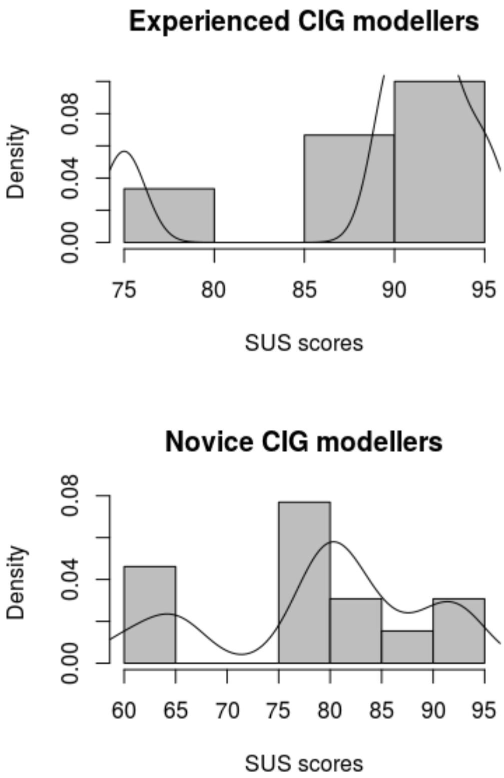


Figure 6.2: Density plots of SUS scores on language constructs usability

I continued to carry out in-depth analyses of the collected data to test the hypotheses that were set out for this study. To start with, recall the first null hypothesis, $\mathbf{H}_0$**:** CIG modellers do not perceive the language constructs of *FCIG* as usable. Table 6.2 presents the statistics that were calculated from the samples of SUS scores on *FCIG* language constructs' usability. The mean and median values of the SUS scores from experienced and novice CIG modellers were both higher than 75. The results indicate that the first null hypothesis can be rejected. This means that there is evidence that CIG modellers perceived the language constructs of *FCIG* as usable. Therefore, we can say that both novice and experienced CIG modellers perceived *FCIG* as a usable CIG modelling language.

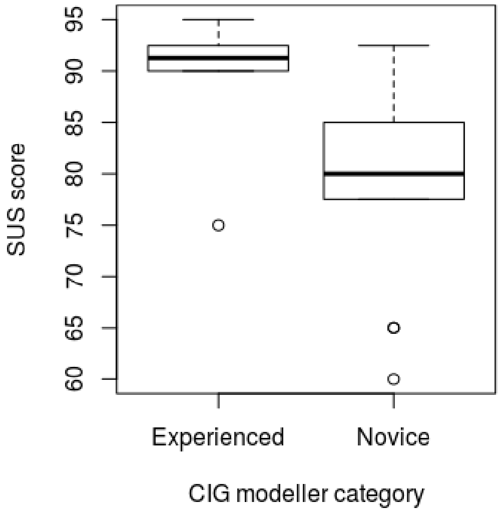


FIGURE 6.3: Box plots of SUS scores on language constructs usability

TABLE 6.3: Shapiro-Wilk test results on SUS scores of language constructs usability

CIG modeller category	W	p-value
Experienced	0.74	0.014
Novice	0.90	0.14

6.3.2.2 SUS score levels between experienced and novice CIG modellers

Thereafter, I continued to test the second null and second alternative hypotheses. I began by using the Shapiro-Wilk test to test the samples of computed SUS scores from the study for normality. Table 6.3 shows the details of the Shapiro-Wilk test results for all two samples which comprised of experienced and novice CIG modellers. The Shapiro-Wilk test result for the sample of SUS scores from experienced CIG modellers had a p-value that was small ($p < 0.05$). Hence, the results showed that the SUS scores from the sample of experienced CIG modellers were not from a normally distributed population. Due to the fact that the SUS scores did not come from a normally distributed population, I used a Wilcoxon Rank Sum test to test for statistical significance of the differences measured from the two conditions of the experiment.

Recall the second null hypothesis, $\mathbf{H}_0$: There is no difference in SUS scores between the ratings from experienced CIG modellers and those from novice CIG modellers. The medians of SUS scores from experienced CIG modellers and novice CIG modellers were 89.17 and 79.23, respectively. A Wilcoxon Rank Sum test was carried out to evaluate the differences in CIG modellers. The test showed that there was no significant effect of CIG modellers (The mean ranks of experienced CIG modellers and novice CIG modellers were 13.67 and 8.31, respectively; U = 0.90, Z = 1.95, p = 0.05, r = 0.45). Due to the fact that the effect of CIG modellers on SUS scoring was not significant, the second null hypothesis can not be rejected. This means that there is no significant difference in SUS scores between the ratings from experienced CIG modellers and those from novice CIG modellers.

Table 6.4: Responses from the SUS survey with experienced CIG modellers

Question	**Score**				
	1	**2**	**3**	**4**	**5**
1 - Would use the grammar frequently	0 – 0%	0 – 0%	2 – 33.3%	2 – 33.3%	2 – 33.3%
2 - Grammar was unnecessarily complex	6 – 100%	0 – 0%	0 – 0%	0 – 0%	0 – 0%
3 - Grammar was easy to use	0 – 0%	0 – 0%	1 – 16.7%	3 – 50%	2 – 33.3%
4 - Need support of a technical person	4 – 66.7%	2 – 33.3%	0 – 0%	0 – 0%	0 – 0%
5 - Functions were well integrated	0 – 0%	0 – 0%	0 – 16.7%	4 – 66.7%	1 – 16.7%
6 - Too much inconsistency	6 – 100%	0 – 0%	0 – 0%	0 – 0%	0 – 0%
7 - Most people would learn quickly	0 – 0%	0 – 0%	1 – 16.7%	0 – 0%	5 – 83.3%
8 - Grammar very cumbersome to use	6 – 100%	0 – 0%	0 – 0%	0 – 0%	0 – 0%
9 - Confident using the grammar	0 – 0%	0 – 0%	0 – 0%	3 – 50%	3 – 50%
10 - Needed to learn a lot of things	5 – 83.3%	0 – 0%	1 – 16.7%	0 – 0%	0 – 0%

6.4 Discussion

Although SUS is a subjective assessment of usability, it can be considered as a valid and reliable indicator of usability [249]. It is argued that the SUS has undergone some considerable amount of psychometric testing to evaluate its validity, reliability and sensitivity [231, 250]. Furthermore, other studies have shown that results from the SUS can reliably converge at samples that are as low as eight [231, 294]. Some researchers have argued that the word *'cumbersome'* in statement number eight of the SUS questionnaire can cause confusion when posed to non-native English speaking participants in a multinational setting [250, 295]. Since this study was carried out in sub-Saharan Africa which is a predominantly non-native English speaking region, the validity of the results in the SUS survey was not negatively affected as the participants in the study were all proficient in the English language. Five experienced CIG modellers were university graduates who had completed their formal education from institutions whose medium of

Table 6.5: Responses from the SUS survey with novice CIG modellers

Question	**Score**				
	1	**2**	**3**	**4**	**5**
1 - Would use the grammar frequently	0 – 0%	0 – 0%	4 – 30.8%	6 – 46.2%	3 – 23.1%
2 - Grammar was unnecessarily complex	8 – 61.5%	4 – 30.8%	1 – 7.7%	0 – 0%	0 – 0%
3 - Grammar was easy to use	0 – 0%	0 – 0%	3 – 23.1%	3 – 23.1%	7 – 53.8%
4 - Need support of a technical person	6 – 46.2%	3 – 23.1%	1 – 7.7%	2 – 15.4%	1 – 7.7%
5 - Functions were well integrated	0 – 0%	0 – 0%	1 – 7.7%	8 – 61.5%	4 – 30.8%
6 - Too much inconsistency	9 – 69.2%	4 – 30.8%	0 – 0%	0 – 0%	0 – 0%
7 - Most people would learn quickly	0 – 0%	3 – 23.1%	2 – 15.4%	4 – 30.8%	4 – 30.8%
8 - Grammar very cumbersome to use	9 – 69.2%	3 – 23.1%	1 – 7.7%	0 – 0%	0 – 0%
9 - Confident using the grammar	1 – 7.7%	1 – 7.7%	4 – 30.8%	3 – 23.1%	4 – 30.8%
10 - Needed to learn a lot of things	4 – 30.8%	8 – 61.5%	1 – 7.7%	0 – 0%	0 – 0%

instruction was English. The sixth CIG modeller was a native English speaker. Furthermore, all novice CIG modellers were university students who had received their prior education in English.

The results in Section 6.3.1 show that both novice and experienced CIG modellers rated the grammar of *FCIG* modelling language highly on the system usability scale. Hence it can be said that the participants in this study perceived the grammar of *FCIG* to be usable. Such a grammar evaluation can provide invaluable insights that can assist a software language designer to improve a software language of interest. Hence contributing to the likelihood of the software language to be adopted in practice.

From the ten dimensions of usability assessment on the SUS scale, the majority of both the novice and expert CIG modellers indicated that *FCIG*'s grammar was easy to use, had well integrated structural elements, did not have too much inconsistency in addition to not being very cumbersome to use. The reason for *FCIG* being perceived as a language with an easy-to-use and consistent grammar might stem from the fact that the language has a small and clear vocabulary consisting of just three main language constructs. Simplicity, clarity and consistency are widely accepted to be essential qualities of good modelling languages [180, 296, 297]. This can be further evidenced by the comments that the participants gave in their qualitative feedback after the survey. The following comments are selected from the feedback that was given by the novice CIG modellers:

1. Participant *p5* commented, *“The three constructs explain the model well. Specifically Condition as it can load data that’s vital to making an informed recommendation”.*

2. Participant *p10* commented, *“It was quite idiomatic and the syntax flows”.*

Most experienced CIG modellers had similar positive perceptions across the same four usability assessment dimensions indicated positively by novice CIG modellers. In addition to the four dimensions, most experienced CIG modellers also indicated that the grammar of the experimental language was not unnecessarily complex. Most of the experienced CIG modellers may have that perception on the grammar because of their prior familiarity with the process of modelling clinical knowledge which is typically carried out using a general purpose programming language. Furthermore, the experienced modellers can have such a positive perception because the semantics and vocabulary of the new CIG modelling language are based on existing concepts that are widely used in the clinical domain. Domain-appropriateness of a language, that entails that a software language be powerful enough to capture major domain concepts and be able to match the mental model of the domain, is very important for software language adoption [227]. This can also be evidenced from the comments that experienced CIG modellers gave in their feedback after the experiment as follows:

1. Participant *p1* commented, *“Condition, action, recommendation. These really cover the basics one would need to use in this kind of setup”.*

2. Participant *p5* commented, *“The syntax, CAR for Conditions, Actions and Recommendations were easy to follow”.*

Although the mean SUS scores for both novice and expert CIG modellers were different, the statistical test results in Section 6.3.1 show that the perception of experienced CIG modellers on the usability of the grammar of *FCIG* was not significantly different to that of novice CIG modellers. This is an expected result because the grammar of *FCIG* uses a small set of language concepts that have a direct mapping to the clinical guideline formalization concepts. By employing a small set of language concepts with adequate expressive power, both novice and experienced modellers are likely to find the grammar as usable. Incompatible domain abstractions in a DSL grammar can introduce limitations that can negatively impact a DSl’s usability [158, 213, 298].

6.5 Chapter summary

In this chapter, I evaluated the usability of *FCIG*'s grammar which is a novel, simple and compact syntax for modelling evolving CIGs. I achieved this by assessing the perceptions of CIG modellers on the usability of *FCIG*'s grammar. Novice CIG modellers were recruited from the University of Cape Town in South Africa where as experienced CIG modellers were recruited from EMR system implementing organisations in Malawi and South Africa. *FCIG* was found to have a pragmatic grammar for modelling CIGs. Both novice and experienced CIG modellers perceived *FCIG*'s grammar as a usable and practical grammar for modelling evolving computer-interpretable guidelines. In the next chapter, I describe how I evaluated *FCIG* for its efficacy in modelling evolving CIGs by comparing it with the HL7 certified standard Arden Syntax.

Chapter 7

Experimental evaluation of *FCIG*

7.1 Introduction

As CPGs evolve over time, their corresponding computer-interpretable guidelines (CIGs) in clinical decision support systems are required to be kept up-to-date so that clinical advice is based on correct guideline recommendations. A CIG modelling architecture that explicitly models the CIG elements that are affected by clinical practice guideline (CPG) changes has the potential to improve the maintenance of CIGs by enabling effective and efficient computational support, not available in current CIG modelling environments, for encoding and maintaining CIGs. In order to enable such computational support, a CIG modelling language, *FCIG*, has been developed for use in a four-layer model-driven architecture. An experimental CIG modelling environment for *FCIG* has been implemented in Eclipse.

Recall the discussion on Arden Syntax in Chapter 2. Arden Syntax was established in 1989 and subsequently developed as a Health Level Seven (HL7) certified standard for modelling computer-interpretable guidelines [73, 79]. HL7 is a not-for-profit, American National Standards Institute (ANSI)-accredited standards developing organisation dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services [73]. Practical and technical limitations have forced researchers developing guideline modelling formalisms and execution software to confine the use of their technology to their home institutions [80]. Arden Syntax represents procedural clinical knowledge in MLMs. Each MLM contains sufficient knowledge to make a single decision that invokes a specific action [78]. Arden syntax has been used for clinical decision support by generating clinical alerts, diagnostic interpretations, management messages and screening for research studies [81].

Arden4Eclipse is an editor for writing Arden Syntax Medical Logic Modules (MLMs) in the Eclipse IDE. It integrates Arden2ByteCode, an open source Arden Syntax compiler, so that Arden Syntax code can be easily written as well as executed. Arden2ByteCode runs on Java Virtual Machines (JVM) and translates Arden Syntax directly to Java bytecode (JBC) executable on JVMs [299]. It also serves as runtime environment for execution of the compiled bytecode. For straightforward use there is an Arden Syntax Editor plugin for the Eclipse IDE which integrates Arden2ByteCode so Arden Syntax code can be written and executed. Unlike *FCIG*, Arden Syntax has a formal syntax but no formal semantics [300].

Significant effort is required to maintain guideline-based clinical decision support systems so that their recommendations are based on up-to-date CPGs. Adoption of a CIG modelling language that uses CPG structural elements that are affected by changes as representation primitives has the potential to provide computational-support for modelling evolving CIGs. This work introduced *FCIG*, in Chapter 5, as such a language that has an explicit specification of elements that are affected by CPG changes in its formal language model.

This chapter presents the methods and related results of an experimental evaluation study that I carried out to compare the novel CIG modelling language *FCIG* with the HL7 standard Arden Syntax.

7.2 Research methods

This section describes the methods that were adopted in this evaluation study.

7.2.1 Research question

Recall the third research question **RQ3** in Section 3.1:

RQ3: *What is the effect of modelling an evolving CIG using FCIG in comparison with the HL7 standard for modelling CIGs?*

In this study, I investigated the effect of modelling CIGs using *FCIG*, a novel language that has fine-grained CPG representation primitives that are affected by CPG changes. I carried out this investigation by comparing *FCIG* with Arden Syntax, the HL7 standard for modelling CIGs. At one-point, a group of novice CIG modellers carried out a set of CIG modelling tasks using *FCIG*. At another point, the same group of CIG modellers carried out the same set of CIG modelling tasks using Arden Syntax. The CIG modelling tasks were counter-balanced in order to reduce order effects. I collected and analysed data from the two groups of activities to measure differences in the following: time spent on the CIG modelling tasks; success rate, number of errors and error rate; efficiency on task; lines of code; and usability perception.

7.2.2 Research design and approach

A mixed methods research design was used. Recall the advantages of a mixed methods research design discussed in Chapter 6. A mixed methods approach combines quantitative and qualitative techniques or methods into a single study [19]. A mixed methods research design can allow a researcher to develop rich insights into various phenomena of interest that can not be fully understood using only a qualitative or quantitative method [293]. In this part of the study, I conducted an experiment with novice CIG modellers. The novice CIG modellers carried out the same set of CIG modelling tasks using both *FCIG* and Arden Syntax. Quantitative data regarding the time taken to complete CIG modelling tasks, the number of tasks completed, the number of errors on completed tasks, lines of code and subjective usability ratings was collected and analysed. In addition, I collected and analysed qualitative data in order to gain a deeper understanding of the CIG modelling experience and other subjective factors that may have affected the perception of modellers when using *FCIG*.

7.2.3 Study setup

Two CIG modelling environments were set up. One modelling environment used the fine-grained CIG modelling language, *FCIG*. The other modelling environment used the existing HL7 standard guideline modelling language, Arden Syntax for MLMs. Arden Syntax was chosen because it is a CIG modelling language that is currently being maintained as an international standard. Furthermore, both languages had the same base of model editing support features as both languages were implemented using the same software language implementation framework, Xtext. Standard smart-editing features, made available through the Xtext software language implementation framework, were available for use in the Eclipse integrated development environment(IDE). The standard smart-editing features that are enabled by Xtext and are provided by the Eclipse IDE out-of-the-box include: syntax highlighting, error checking, auto-completion, content-assist, formatting, quick-fixes and outline view [207]. Both modelling environments were setup on the same computer with the following specification:

- Computer type: Dell Proline desktop computer with mouse and keyboard
- Monitor: Samsung 20 inch high-definition multimedia interface (HDMI) colour monitor
- Memory: 3.8 gigabytes (GB)
- Processor: Intel Core i3-2100 with 3.1 gigahertz (GHz) processing speed
- Operating system: Ubuntu 64-bit 16.04 long-term support (LTS)
- Disk capacity: 156 GB
- Web browser: Mozilla Firefox for Ubuntu 52.0.1
- Integrated Development Environment (IDE): Eclipse Neon.

7.2.4 Study participants

This section describes the characteristics of the participants that took place in the experiments.

Participation in this experiment was open to individuals with basic software engineering skills similar to that required for entry level positions in clinical decision support systems development roles. Such entry level positions in countries like Malawi are usually carried out by fresh graduates with a diploma or a degree in computing with basic software engineering knowledge that includes experience with computer programming using multiple software languages and IDEs. To that regard, computer science students in third year of study and above from the University of Cape Town participated in the study as novice modellers. By third year of university education, the participants will have acquired the basic software engineering skills required for entry level clinical decision support systems design and development roles in countries like Malawi. Participants were given a small monetary incentive of ZAR50.00 in appreciation of their time. Figure 7.1 shows one of the study participants carrying out computer-interpretable guideline modelling tasks during the experiment.

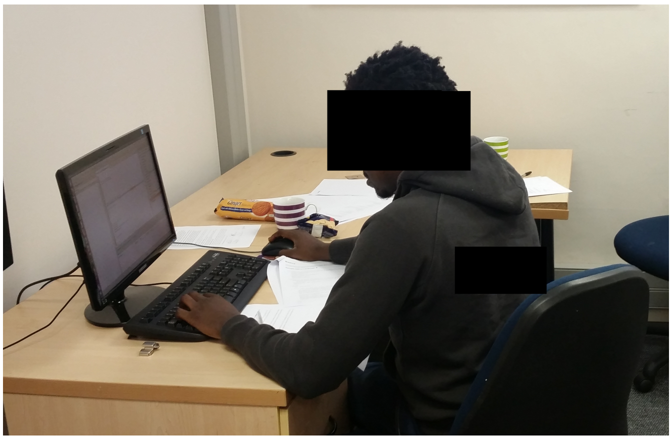

[redacted] [redacted]

FIGURE 7.1: A study participant taking part in the experiment

7.2.5 Experiment design

There were two experimental conditions for the experiment. The first, *Condition A*, was where participants carried out tasks using the HL7 standard, Arden Syntax. The second, *Condition B*, was where participants carried out tasks using *FCIG*.

In the experiment, a repeated measures experimental design was used. A within-subjects or repeated measures design uses the same subjects with every condition of the research, by manipulating the explanatory variable on the same subjects [301, 302]. As such, each participant was exposed to both experimental conditions, *Condition A* and *Condition B*. This technique attempted to reduce the effects of natural variation between participants upon the results.

Practice effects may occur in an experiment, causing participants to become better at a task over time. In addition, fatigue effects may also occur causing participants to become worse at a task through boredom and fatigue [302]. In order to reduce the order effects that may have occurred during an experiment, counterbalancing using a crossover design was used in the design of the experiments. Counterbalancing is a technique whereby each of the two groups performs all tasks but in reverse order to each other to offset practice or fatigue effects [302]. The research study participants were randomly assigned to one of two groups. The first group, *Group one*, subjected participants to experimental *Condition A* followed by experimental *Condition B*. The second group, *Group two*, subjected participants to experimental *Condition B* followed by experimental *Condition A*. Table 7.1 shows the groups and the sequence in which the participants were subjected to both experimental conditions.

A pretest-posttest design was used in the study. A pretest-posttest experiment design ensures that the research design is a true experimental design so that the degree of change occurring because of particular research conditions can be measured. Figure 7.2

Table 7.1: Counterbalancing in the experiments

Category	**First Condition**	**Second Condition**
Group one	*Condition A*	*Condition B*
Group two	*Condition B*	*Condition A*

shows how the pretest-posttest design with counterbalancing technique was used in the research design.

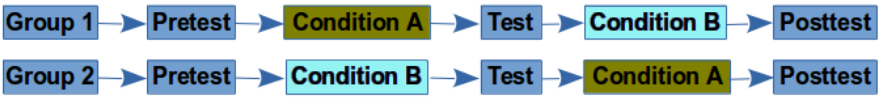


Figure 7.2: Pretest-posttest design

The pre-test for the experiment was that only those participants that had at least had basic computer programming skills and had no prior CIG modelling experience were eligible to enrol into the study. The post-test was the performance comparison between the experimental *Condition A* and experimental *Condition B* collective measurements in terms of success rate, time taken to complete the tasks, error rate, lines of code, and usability perception ratings measured during the CIG modelling tasks.

7.2.5.1 Guideline modelling tasks

The guideline modelling tasks simulated guideline encoding and maintenance activities that are typically undertaken when a CIG modeller creates new or revises existing CIGs. The tasks were validated and verified by two experienced CIG modellers from Malawi. Each participant was asked to encode three guideline recommendations from the CPG for managing ART in special situations extracted from the 2011 Malawi HIV management guidelines. Only three guideline recommendations were selected for the first task to reduce fatigue effects in the experiment. Thereafter, participants were asked to update a previously coded CPG for managing ART in special situations from the 2011 Malawi HIV management guidelines to its 2014 version. The conveniently selected CPG was the one that included all of the ten types of CPG changes [2] to simulate adequate CPG version update activities. The tasks were as follows:

1. Encode three new guideline recommendations from the 2011 CPG for managing ART in special situations
2. Update the complete 2011 CPG for managing ART in special situations to its 2014 version

Task one was made up of three subtasks and task two was made up of 10 subtasks. The details of the subtasks for the two tasks are available in Appendix G.

7.2.5.2 Research protocol

Each participant went through the experiment independently of other study participants. All participants used the same computer at different times in a secluded room in order to reduce design contamination for the experiments and to enhance my ability to carry out effective observations. Each participant followed the following procedure:

1. The purpose of the experiment was explained to the participant
2. The participant completed an informed consent form. Refer to Section H.1 in Appendix H for more details
3. The procedure of the experiment was explained to the participant. Refer to Section H.2 in Appendix H for more a script that was read to each participant
4. The participant was randomly assigned to either experimental *Group one* or *Group two* where language A was Arden Syntax and *FCIG* respectively
    - The experiment administrator assigned a sealed and unmarked envelope containing written instructions for the experiment. The instructions were previously randomly set for either *Group one* or *Group two* through a digital randomisation application called Random UX[1].
    - There were two sets of instructions for the experiment. The first set of instructions was for participants in *Group one*[2] and the second set of instructions was for participants in *Group two*[3]
5. The participant was allocated a preset computer
6. The participant followed appropriate instructions for *Group one* or *Group two* depending on the group allocated to them
7. The participant undertook self-paced basic training for language A using an appropriate tutorial for *Group one*[4] or *Group two*[5] accordingly

[1] https://play.google.com/store/apps/details?id=ru.uxapps.random&hl=en

[2] https://github.com/yamiko/CIG-Modelling-Languages/wiki

[3] https://github.com/yamiko/cig_modelling_languages/wiki

[4] https://github.com/yamiko/CIG-Modelling-Languages/wiki/Language-A

[5] https://github.com/yamiko/cig_modelling_languages/wiki/Language-A

8. The participant was issued with an appropriate set of instructions for the CIG modelling tasks on a printout depending on the sequence of the experimental conditions. Refer to Section H.3 and Section H.4 in Appendix H for *Group one* and *Group two* instructions respectively

9. The participant was observed whilst carrying out tasks for the experiment

10. The participant carried out the tasks using a guideline modelling language according to the group one was allocated to in step three

11. The participant was asked to fill an online post-test questionnaire. Refer to Section H.5 in Appendix H for a list of questions asked in the post-test questionnaire

12. The participant was given a five minute break

13. The participant undertook self-paced basic training for language B using an appropriate tutorial for *Group one*[6] or *Group two*[7] accordingly

14. The participant carried out the same set of guideline modelling tasks as those issued in step six using the CIG modelling language allocated in step 11. Refer to Section H.3 and Section H.4 in Appendix H for *Group one* and *Group two* instructions respectively

15. The participant was asked to fill an online post-test questionnaire after completing tasks with the second experimental condition. Refer to Section H.5 in Appendix H for more details on the set of questions asked in the post-test questionnaire

16. The participant was interviewed to get a qualitative description of the CIG modelling language that the participant preferred

17. The participant signed in the participant log and received ZAR50.00 to compensate for their time.

7.2.6 Data collection methods

I used the following data collection methods in the study:

- Low level computer logs from the Fluorite plugin [303] of the Eclipse IDE were used to collect data on guideline modellers' interaction with the guideline modelling environments.

[6] https://github.com/yamiko/CIG-Modelling-Languages/wiki/Language-B

[7] https://github.com/yamiko/cig_modelling_languages/wiki/Language-B

- Questionnaires were used. Where possible, online questionnaires were deployed for data collection. In addition, the SUS that uses a Likert scale was used as a standardised questionnaire to collect data about participants' perceptions on a particular CIG modelling language.

- Observations were used to record details as modellers were carrying out the CIG modelling tasks. I observed from close enough a distance to see what was happening but not so close as to interfere with the CIG modelling tasks.

7.2.7 Criteria to address the research question RQ2

In order to address research question **RQ2**, I compared collective performance measurements from *FCIG* modellers and Arden Syntax modellers. Each participant attempted to use the benchmark and HL7 standard guideline modelling language, Arden Syntax, as experimental *Condition A*. And at another point each of the participants used the new fine-grained CIG modelling language, *FCIG*, as experimental *Condition B*. Hence the independent variable was the particular CIG modelling language used during the experiment. Participants were randomly divided into two groups, *Group one* and *Group two*, as categorised in Section 7.2.5. As each participant was subjected to perform the same set of tasks using both CIG modelling languages, data was collected to measure: task time, success rate, error rate and efficiency. Success rate, task time and errors are fundamental metrics that can be applied to scientific or user research [231]. In order to measure and evaluate these various dimensions separately, sub-questions are devised for each of the dimensions as discussed in the following sub-sections.

7.2.7.1 Task time

In order to measure and evaluate task time, I asked the following sub-question:

SRQ1: *What is the effect of using FCIG, in comparison with Arden Syntax, on task time?*

Task time, which is defined as how long a user spends on an activity, can be reported as an average of time spent on tasks per participant [231]. Task time is a fundamental evaluation metric that can assist in determining the quality of user experience [233]. Task time can be related to efficiency and effectiveness of an artefact which are fundamental goals of usability [231, 233, 304]. Task time was measured as the duration between the time a participant started and ended a guideline modelling task. However, some guideline modelling tasks might have been attempted but not completed whilst others could have been completed. Task time can be measured in the following ways: 1) Task completion time: time of participants who complete the task successfully, 2) Time until failure: time on task until participant gives up on uncompleted tasks and 3) Total time on task: the total duration of time users spend on a task for both completed and uncompleted tasks [231]. In order to address sub-question **SRQ1**, data regarding time on task between the *FCIG* modellers and Arden Syntax modellers was compared. In order to guide the comparison, three alternative hypotheses were derived in line with the different ways that time can be measured.

I formulated the following hypothesis for time taken on completed tasks:

H$_0$: There is no difference in the mean time taken on completed tasks between *FCIG* modellers and Arden Syntax modellers.

H$_1$: The mean time taken on completed tasks by *FCIG* modellers is less than the mean time on completed tasks by Arden Syntax modellers.

In addition, I formulated the following hypothesis for time taken on uncompleted tasks:

H$_0$: There is no difference in the mean time taken on uncompleted tasks between *FCIG* modellers and Arden Syntax modellers.

H$_1$: The mean time taken on uncompleted tasks by *FCIG* modellers is less than the mean time on uncompleted tasks by Arden Syntax modellers.

At the same time, I also formulated the following hypothesis for total time taken on tasks, that included time taken on both completed and uncompleted tasks:

H$_0$: There is no difference in the total time taken on tasks between *FCIG* modellers and Arden Syntax modellers.

H$_1$: The total time taken on tasks by *FCIG* modellers is less than the total time taken on tasks by Arden Syntax modellers.

Noting that the nature of specific subtasks can influence the total time a CIG modeller spends on a particular task, I formulated the following hypothesis for time taken on subtasks:

H$_0$: There is no significant difference in the time taken on subtasks between *FCIG* modellers and Arden Syntax modellers.

H$_1$: The time taken on subtasks by *FCIG* modellers is less than the time taken on subtasks by Arden Syntax modellers.

I used the *Shapiro-Wilk* test to test for normality and to determine a relevant statistical test to test for statistical significance.

7.2.7.2 Success rate

In order to measure and evaluate success rate, I posed the following sub-question:

SRQ2: *What is the effect of using FCIG, in comparison with Arden Syntax, on success rate?*

Success rates also called completion rates, typically collected as a binary measure of task success or task failure, are the most fundamental of usability metrics [231]. Success rate is also a key evaluation metric of usability that can assist in determining the quality of user experience [231, 233]. In order to address the sub-question, data on success rates was compared between *FCIG* modellers and Arden Syntax modellers. Some guideline modelling tasks might have been completed by the participants whilst other tasks might not have been completed. Each task was marked with a binary measure to indicated whether the task was completed or not.

The following hypothesis was used on success rates:

H_0: There is no difference in success rates between *FCIG* modellers and Arden Syntax modellers.

H_1: The success rate of Arden Syntax modellers is more than the success rate of *FCIG* modellers.

I used a Chi-squared test to test for statistical significance of the differences in success rates between *FCIG* modellers and Arden Syntax modellers.

7.2.7.3 Number of errors and error rates

In order to measure and evaluate the number of errors and the error rate, I devised the following sub-questions:

SRQ3A: *What is the effect of using FCIG, in comparison with Arden Syntax, on error rate?*

SRQ3B: *What is the effect of using FCIG, in comparison with Arden Syntax, on the number of errors made?*

An error, typically analysed as a binary measure, is any unintended action, slip, mistake, or omission a user makes while attempting a task [231].Number of errors on task is also a key evaluation metric of usability that can assist in determining the quality of user experience [231, 233]. In this experiment, I considered all syntax and logical errors that were evident in the source code that was saved after each CIG modelling task. I loaded the source code files for each task and for each participant in the Eclipse Integrated Development Environment (IDE) and counted the number of syntax errors that were reported by the IDE. I further manually validated all conditions and actions for each guideline in each source code file for the presence of semantic errors. In order to address sub-question **SRQ3A**, binary measures on errors between *FCIG* modellers and Arden Syntax modellers were compared. The following hypothesis was used on error rates:

H_0: There is no difference in the error rate between *FCIG* modellers and Arden Syntax modellers.

H_1: The error rate of the *FCIG* modellers is less than the error rate made by Arden Syntax modellers.

Since I collected data on the occurrence of errors as binary measures and that I adopted a repeated measures design for the experiments, I used the McNemar's Chi-squared test with continuity correction to test for statistical significance of any differences between experimental *Condition A* and experimental *Condition B* error rates.

In order to address sub-question **SRQ3B**, the number of errors between *FCIG* modellers and Arden Syntax modellers were compared. The following hypothesis was used on number of errors on tasks:

H_0: There is no difference in the number of errors between *FCIG* modellers and Arden Syntax modellers.

H_1: The number of errors made in *FCIG* modellers is less than the number of errors made in Arden Syntax modellers.

On the number of errors on subtasks, I used the following hypothesis:

H_0: There is no difference in the number of errors made on subtasks between *FCIG* modellers and Arden Syntax modellers.

H_1: The number of errors made on subtasks between experimental *Condition A* is significantly higher than the number of errors on subtasks in *FCIG* modellers.

I used the *Shapiro-Wilk* test to test for normality and to determine a relevant statistical test to test for statistical significance of any differences on number of errors between *FCIG* modellers and Arden Syntax modellers.

7.2.7.4 Efficiency

In order to measure and evaluate efficiency, I asked the following sub-question:

SRQ4: *What is the effect of using FCIG on efficiency?*

Efficiency, indicated as task completion time, is the relation between the accuracy and completeness with which users achieve certain goals and the resources expended in achieving them [305]. Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources as a ratio between task success rate and task time [306]. The formula for calculating efficiency is presented below:

$$\frac{task\ success\ rate}{total\ task\ time} \tag{7.1}$$

Efficiency is a fundamental goal of usability of an artefact [231, 233]. In order to address this sub-question, task success rate to task time ratios between the *FCIG* modellers and Arden Syntax modellers were compared. The following hypothesis was used on efficiency:

H$_0$: There is no difference in the task success rate to task time ratio between *FCIG* modellers and Arden Syntax modellers.

H$_1$: The task success rate to task time ratio of the *FCIG* modellers is more than the task success rate to task time ratio of the Arden Syntax modellers.

I used the Shapiro-Wilk test to test for normality of the samples and its results determined a relevant statistical test that was used to test for statistical significance.

7.2.7.5 Lines of code

In order to measure and evaluate the differences in lines of code (LOC) and effective lines of code (ELOC), I asked the following sub-questions:

SRQ5A: *What is the effect of using FCIG on LOC?*

SRQ5B: *What is the effect of using FCIG on ELOC?*

LOC is a software size metric that that represent the number of lines of source code where as ELOC captures the number of source code lines that exclude comments and blank lines [307]. In order to address these sub-questions, LOC and ELOC measures between the *FCIG* modellers and Arden Syntax modellers were compared. In order to address sub-research question **SRQ5A**, the following hypothesis was used on LOC:

H$_0$: There is no difference in LOC between *FCIG* modellers and Arden Syntax modellers.

H$_1$: The LOC in *FCIG* modellers are less than the LOC of the Arden Syntax modellers.

In order to address sub-research question **SRQ5A**, the following hypothesis was used on ELOC:

H_0: There is no difference in ELOC between *FCIG* modellers and Arden Syntax modellers.

H_1: The ELOC in *FCIG* modellers are less than the ELOC of the Arden Syntax modellers.

I used the Shapiro-Wilk test to test for normality of the samples and its results determined a relevant statistical test that was used to test for statistical significance.

7.2.7.6 SUS scores

In order to measure and evaluate usability perception from SUS ratings scales, I asked the following sub-question:

SRQ6: *What is the effect of using FCIG on SUS scores?*

To reiterate, the system usability scale (SUS) uses a five-point likert scale as a standardised questionnaire to collect data about participants' perceptions on usability [248]. The SUS has undergone some considerable amount of psychometric evaluation that can attest to its validity, reliability and sensitivity [231, 250]. In addition, other studies have shown that results from the SUS can reliably converge at relatively lower samples than other standardised usability questionnaires by converging at samples that are as low as eight [231, 294].

In this study, SUS scores were calculated from the questionnaires that were filled by each participant on *FCIG* modellers and Arden Syntax modellers. In order to address the sub-question on SUS scores, the SUS scores that were calculated from *FCIG* modellers and Arden Syntax modellers were compared. The following hypothesis was used on the calculated SUS scores:

H_0: There is no difference in SUS scores between *FCIG* modellers and Arden Syntax modellers.

H_1: The SUS scores from *FCIG* modellers are higher than the SUS scores from the Arden Syntax modellers.

I used the Shapiro-Wilk test to test for normality and its results determined a relevant statistical test that was used to test for statistical significance.

7.2.7.7 Qualitative data and user perceptions

I collected qualitative data through questionnaires. I carried out a thematic analysis on the collected qualitative data in order to systematically identify relevant patterns. Thematic analysis is a common form of qualitative analysis that records information that is linked by a common idea allowing the researcher to categorise the information into a framework of thematic ideas about it [308]. In this study, I followed the widely used six-step approach to thematic analysis [309]. I started by familiarising myself with the data. Thereafter, I coded interesting features of the data. Using the coded features, I searched for relevant themes by comparing the coded features and grouping all information that was found as related under one theme. Thereafter, I reviewed the themes to ensure that each set of coded data and its related datasets correlated with their respective themes. I further refined the themes by assigning appropriate names and clear definitions of the themes. Lastly, I wrote-up an analytical narrative of the results in relation to the research question and existing literature. The steps in a thematic analysis process do not follow a linear pattern but rather a recursive one [309]. Figure 7.3 illustrates the approach to thematic analysis that I adopted for this study.

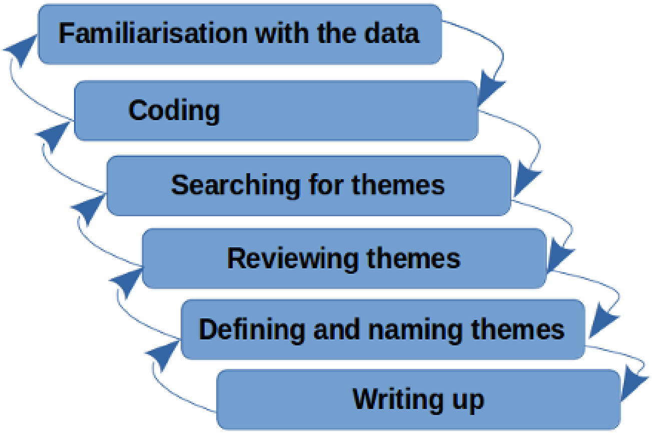


FIGURE 7.3: Thematic analysis approach

7.2.8 Other experimental design validity considerations

The CIG modelling tasks for the experiments were designed in such a way that takes account of experimental mortality and maturation. By having a few tasks that are of reasonable length for both experimental conditions, the likelihood of participants dropping off the experiment before completion was reduced. In addition, the likelihood of participants becoming significantly better at tasks or significantly developing fatigue through excessive repetition of tasks was reduced.

Table 7.2: General purpose programming languages known by novice CIG modellers

Programming language	**Number of participants**
C	3
C++	24
C#	2
Java	25
Python	25

All experiment procedures and instructions were written down before the experiments began so that all study participants are exposed to uniform instrumentation. In addition, all study participants used the same computer at different times. The experimental conditions were not disclosed to study participants to further reduce chances of design contamination.

All study participants were compensated equally for their time in taking part in the experiments. This was aimed at ensuring that there was no compensatory rivalry among study participants or that performance in a specific task was motivated by the amount of compensation.

7.3 Discussion of results

In the subsections that follow, I present and discuss the results from the experiment that I carried out with novice CIG modellers. To repeat, I carried out this experiment to objectively compare *FCIG* against the HL7 standard, Arden Syntax. The raw data from these experiments is in Appendix L.

7.3.1 Participants

There were 26 students from the University of Cape Town that participated in the study. Nineteen of the students were in their fourth year of study; whilst three of the students were undertaking their master's degree studies; and the other four students were undertaking their doctoral degree studies. All the participants had not worked with medical computing systems prior to the study. In addition, all the participants were studying computer science and had working knowledge of at least two general purpose programming languages. Table 7.2 shows the programming language proficiency amongst the participants as reported during the experiments.

7.3.2 Task time

Firstly, recall the hypothesis for task time on completed tasks:

H$_0$: There is no difference in the mean time taken on completed tasks between *FCIG* modellers and Arden Syntax modellers. I formulated this as the first null hypothesis.

H$_1$: The mean time taken on completed tasks by *FCIG* modellers is less than the mean time on completed tasks by Arden Syntax modellers. I formulated this as the first alternative hypothesis.

Secondly, recall the hypothesis for time taken on uncompleted tasks:

H$_0$: There is no difference in the mean time taken on uncompleted tasks between *FCIG* modellers and Arden Syntax modellers. I formulated this as the second null hypothesis.

H$_1$: The mean time taken on uncompleted tasks by *FCIG* modellers is less than the mean time on completed tasks by Arden Syntax modellers. I formulated this as the second alternative hypothesis.

Thirdly, recall the hypothesis for total time taken on tasks:

H$_0$: There is no difference in the total time taken on tasks between *FCIG* modellers and Arden Syntax modellers. I formulated this as the third null hypothesis.

H$_1$: The total time taken on tasks by *FCIG* modellers is less than the total time taken on tasks by Arden Syntax modellers. I formulated this as the third alternative hypothesis.

Lastly, recall the hypothesis for time taken on subtasks:

H$_0$: There is no difference in the time taken on subtasks between *FCIG* modellers and Arden Syntax modellers. I formulated this as the fourth null hypothesis.

H$_1$: The time taken on subtasks by *FCIG* modellers is less than the time taken on subtasks by Arden Syntax modellers. I formulated this as the fourth alternative hypothesis.

7.3.2.1 Time taken on completed tasks

I started by carrying out some exploratory analysis of the data on mean time taken on completed tasks from the experiment. I created box plots showing the time taken on completed tasks. Thereafter, I checked and validated all odd data points. Figure 7.4 shows the box plots for both task one and two that show the distribution of mean time taken on completed tasks. As part of the exploratory analysis, I created density plots so I could visualise the data. Figure 7.5 shows the density plots for mean time taken

Table 7.3: Shapiro-Wilk test results on mean time taken on completed tasks

Language	**Task**	**W**	**p-value**
Arden Syntax	Task one	0.89	0.001
	Task two	0.82	<0.001
FCIG	Task one	0.93	0.089
	Task two	0.55	<0.001

task one and task two when completed with both modelling languages Arden Syntax and *FCIG*. The initial analysis showed that the samples of mean time taken on tasks might not have come from normally distributed populations.

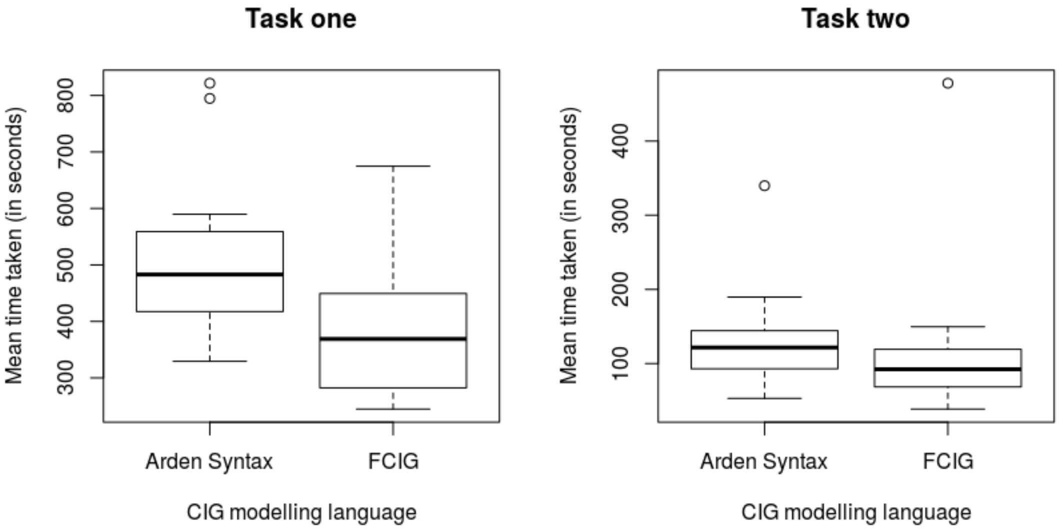


Figure 7.4: Box plots showing mean time taken on completed tasks in the experiment

I continued to test the samples for normality by using the Shapiro-Wilk test on the samples of mean task times recorded in the experiment. The Shapiro-Wilk test results had p-values of less than 0.05 ($p < 0.05$) for the estimates from *FCIG* modellers and Arden Syntax modellers. Table 7.3 shows the details of the Shapiro-Wilk test results. This meant that the samples of mean task times of completed tasks were not from normally distributed populations. Due to the mean task times of completed tasks not coming from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used a Wilcoxon Signed-Rank test to test for statistical significance of the differences measured in the different conditions in the experiment. Since the tests were conducted on both task one and two, the Bonferonni correction was used to account for multiple hypothesis testing.

Recall the first null hypothesis, $\mathbf{H}_0$: There is no difference in the mean time taken on completed tasks between *FCIG* modellers and Arden Syntax modellers. The medians

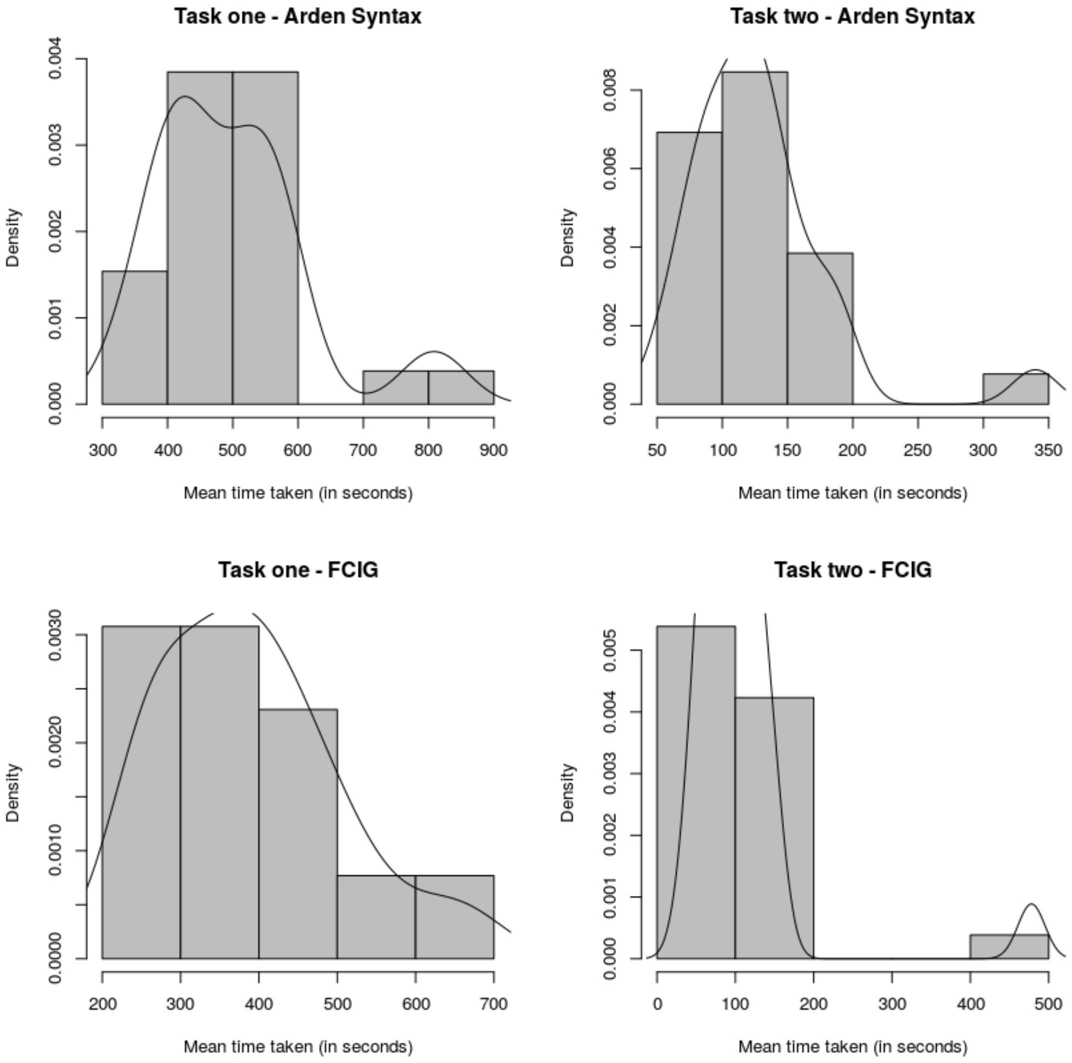


Figure 7.5: Density plots showing time taken on completed tasks in the experiment

of mean time taken on task one for *FCIG* and Arden Syntax were 368.8 and 483 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of mean time taken on task one (W = 44, Z = 3.34, $p < 0.001$ with Bonferonni correction, r = 0.46). The medians of mean time taken on task two for *FCIG* and Arden Syntax were 92.3 and 121.7 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of mean time taken on task two (W = 74, Z = 2.58, $p = 0.017$ with Bonferonni correction, r = 0.36). The statistical test results provide enough evidence such that the first null hypothesis could be rejected. This meant that the first alternative hypothesis could be accepted. The results suggest that a particular kind of CIG modelling language does have an effect on the average time it takes to complete a task. Specifically, the results suggest that tasks are carried out in less time using *FCIG* in comparison with Arden Syntax.

7.3.2.2 Time taken on uncompleted tasks

Recall the second null hypothesis, $\mathbf{H}_0$: There is no difference in the mean time taken on uncompleted tasks between *FCIG* modellers and Arden Syntax modellers. I proceeded with the analysis by evaluating the second null hypothesis. All tasks that were requested during the experiment were completed by all participants. Therefore, the second null hypothesis could not be rejected. The results suggest that there was not enough evidence that pointed to the fact that a particular CIG modelling language has an effect on the ability of a modeller to complete a task.

7.3.2.3 Total time taken on tasks

I carried out some exploratory data analysis on total task times for both task one and task two that were recorded during the experiment. I created box plots showing the total time taken on tasks. I subsequently checked and validated all odd data points. Figure 7.8 depicts the box plots that show the distribution of total time taken on task one and two. I further created density plots in order to visualise the distribution of the samples that were collected during the experiment. Figure 7.7 shows the density plots for total time spent on task one and task two when completed with both modelling languages Arden Syntax and *FCIG*. The exploratory analysis showed that the samples might not have come from a normally distributed population.

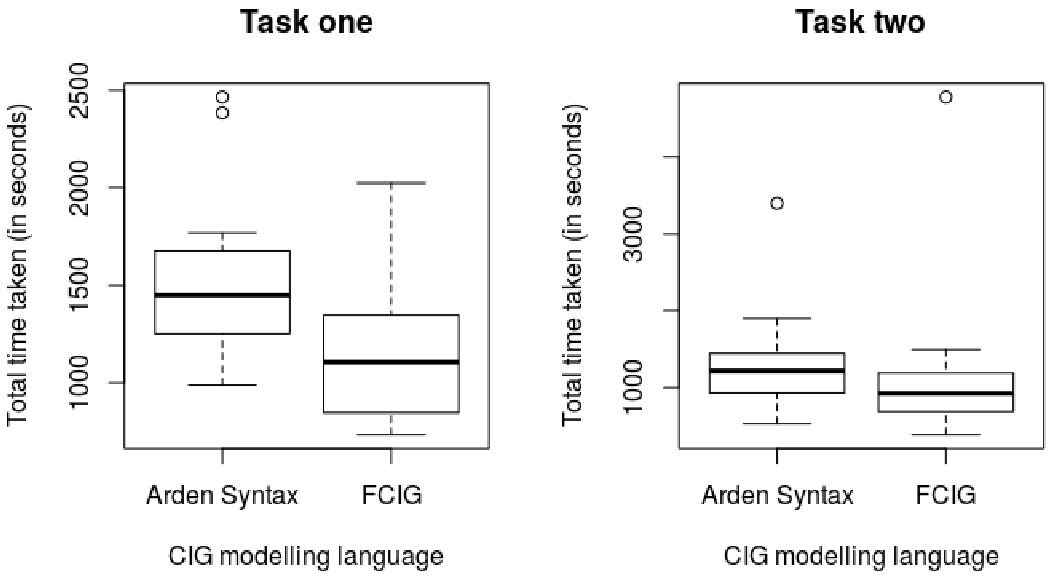


FIGURE 7.6: Box plots showing total time taken on tasks in the experiment

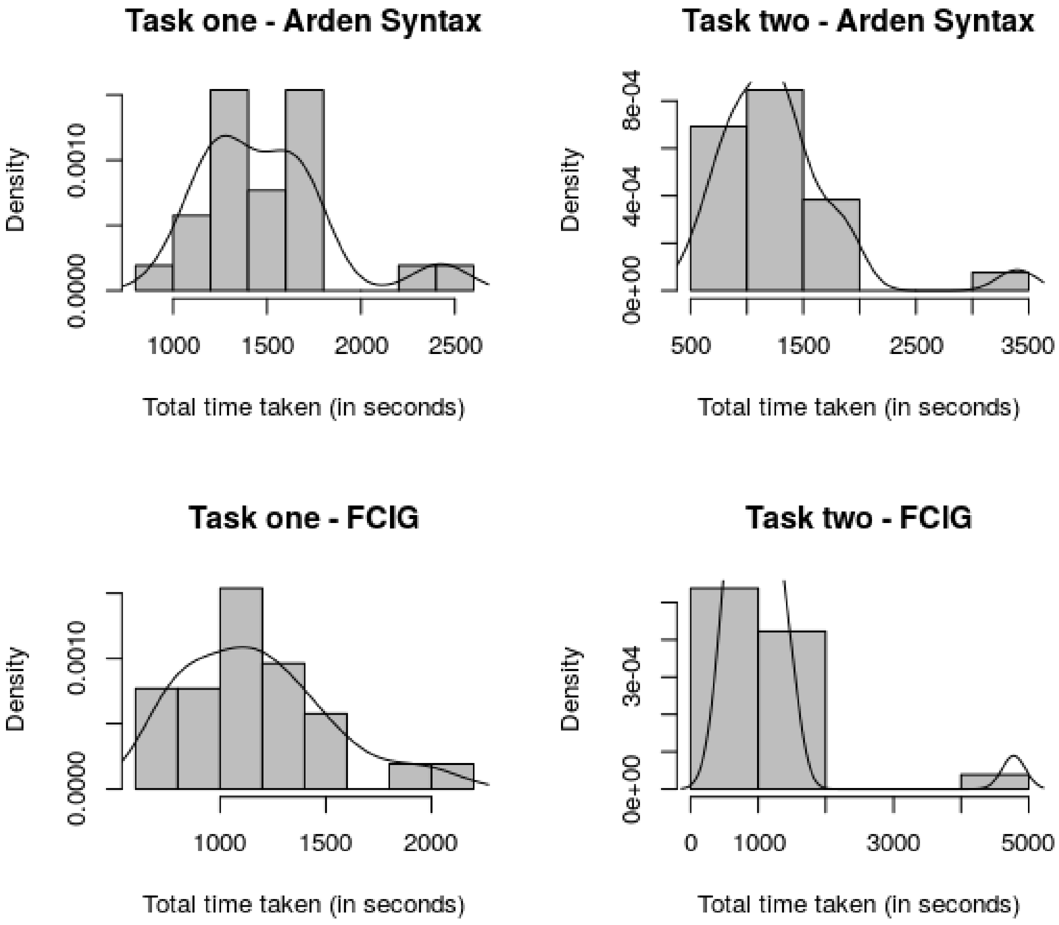


FIGURE 7.7: Density plots showing total time taken on tasks in the experiment

I proceeded to test the samples of total time taken on task for normality using the Shapiro-Wilk test. The Shapiro-Wilk test resulted in p-values that were less than 0.05 ($p < 0.05$) for the estimates from *FCIG* modellers and Arden Syntax modellers. Table 7.4 shows the details of the Shapiro-Wilk test results. For this reason, the samples of total task times for both task one and task two carried out using both Arden Syntax and *FCIG* did not come from normally distributed populations. Due to the total task times not coming from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used a Wilcoxon Signed-Rank test to test for differences and statistical significance of the measures that were recorded for the different conditions in the experiment. I also used the Bonferronni correction to account for multiple hypothesis testing.

Recall the third null hypothesis, $\mathbf{H}_0$: There is no difference in the total time taken on tasks between *FCIG* modellers and Arden Syntax modellers. The medians of total time taken on task one for *FCIG* and Arden Syntax were 1106.5 and 1449 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of total time taken

Table 7.4: Shapiro-Wilk test results on total time taken on tasks

Language	**Task**	**W**	**p-value**
Arden Syntax	Task one	0.89	0.008
	Task two	0.82	<0.001
FCIG	Task one	0.93	0.089
	Task two	0.55	<0.001

on task one (W = 44, Z = 3.34, $p < 0.001$ with Bonferonni correction, r = 0.46). The medians of total time taken on task two for *FCIG* and Arden Syntax were 923 and 1216.5 respectively. A Wilcoxon Signed-rank test showed that there was also a significant effect of total time taken on task two (W = 74, Z = 2.58, $p = 0.017$ with Bonferonni correction, r = 0.36). The statistical results provided enough evidence such that the third null hypothesis could be rejected. This meant that the third alternative hypothesis could be accepted. The results suggest that a particular kind of CIG modelling language does have an effect on the overall time a modeller spends on a task. Specifically, the results suggest that tasks are carried out in less time using *FCIG* in comparison with Arden Syntax.

7.3.2.4 Time taken on subtasks

Since there were statistically significant differences in the total times taken for each of the tasks in the experiment, I continued to analyse the observed differences in each of the subtasks within each task. I started by carrying out some exploratory data analysis on individual subtask times for each task. I created box plots showing the time taken on individual subtasks and subsequently checked and validated all odd data points. Figure 7.8 and Figure 7.9 contain the box plots, that show the distribution of time taken, for the subtasks in task one and two respectively. In addition, I created density plots to visually inspect the distribution of the samples in this analysis. Figure 7.10 shows density plots for task one when completed with both modelling languages Arden Syntax and *FCIG*. Figures 7.11 and 7.12 further depict the density plots for task two when completed with the CIG modelling languages Arden Syntax and *FCIG* respectively. The exploratory analysis showed that most of the samples might not have come from normally distributed populations.

Thereafter, I continued with to test for normality using the Shapiro-Wilk test to test on task times recorded for each subtask in the experiment. The Shapiro-Wilk test results indicated that 84.61% of p-values were below 0.05 ($p < 0.05$). For this reason, the results show that the majority of subtask times in both task one and task two carried out using both Arden Syntax and *FCIG* did not come from normally distributed populations.

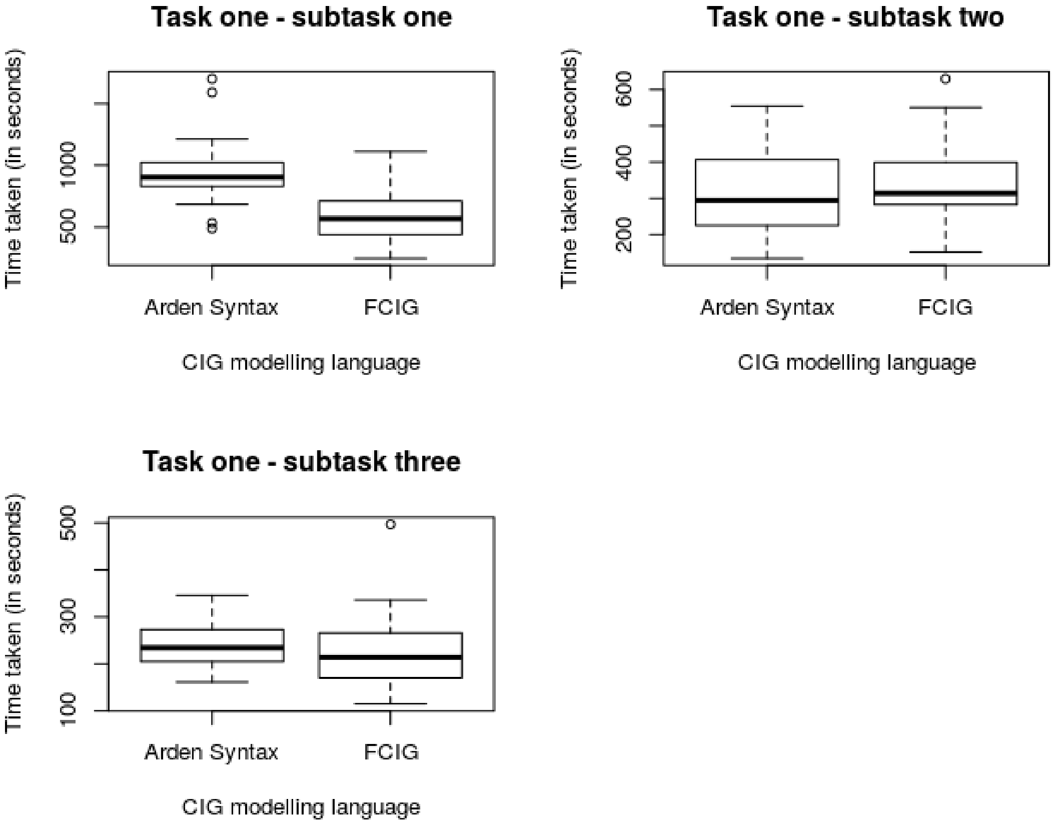


FIGURE 7.8: Box plots showing time taken on each of the subtasks for task one in the experiment

Table 7.5 shows the details of the Shapiro-Wilk test results. Due to the majority of task times not coming from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used the Wilcoxon Signed-rank test to test for statistical significance of the differences measured between subtask times for the two conditions of the experiment.

Recall the fourth null hypothesis, $\mathbf{H}_0$: There is no difference in the time taken on subtasks between *FCIG* modellers and Arden Syntax modellers. The medians of the time taken on each of the subtasks are presented in Table 7.6. The Wilcoxon Signed-rank test results varied. To start with, the statistical test results showed that there was a significant effect of time taken on subtask for subtask one of task one and subtasks four, five, six, seven, and eight of task two. Table 7.7 details the test results on subtask times. The statistical test results provided enough evidence such that the fourth null hypothesis could be rejected on specific subtasks. A closer inspection of these subtasks revealed that these were the tasks that required identification and reuse of fine-grained clinical practice guideline components such as Variable Values. As a result, the fourth alternative hypothesis could be accepted on such kind of tasks. The results suggest

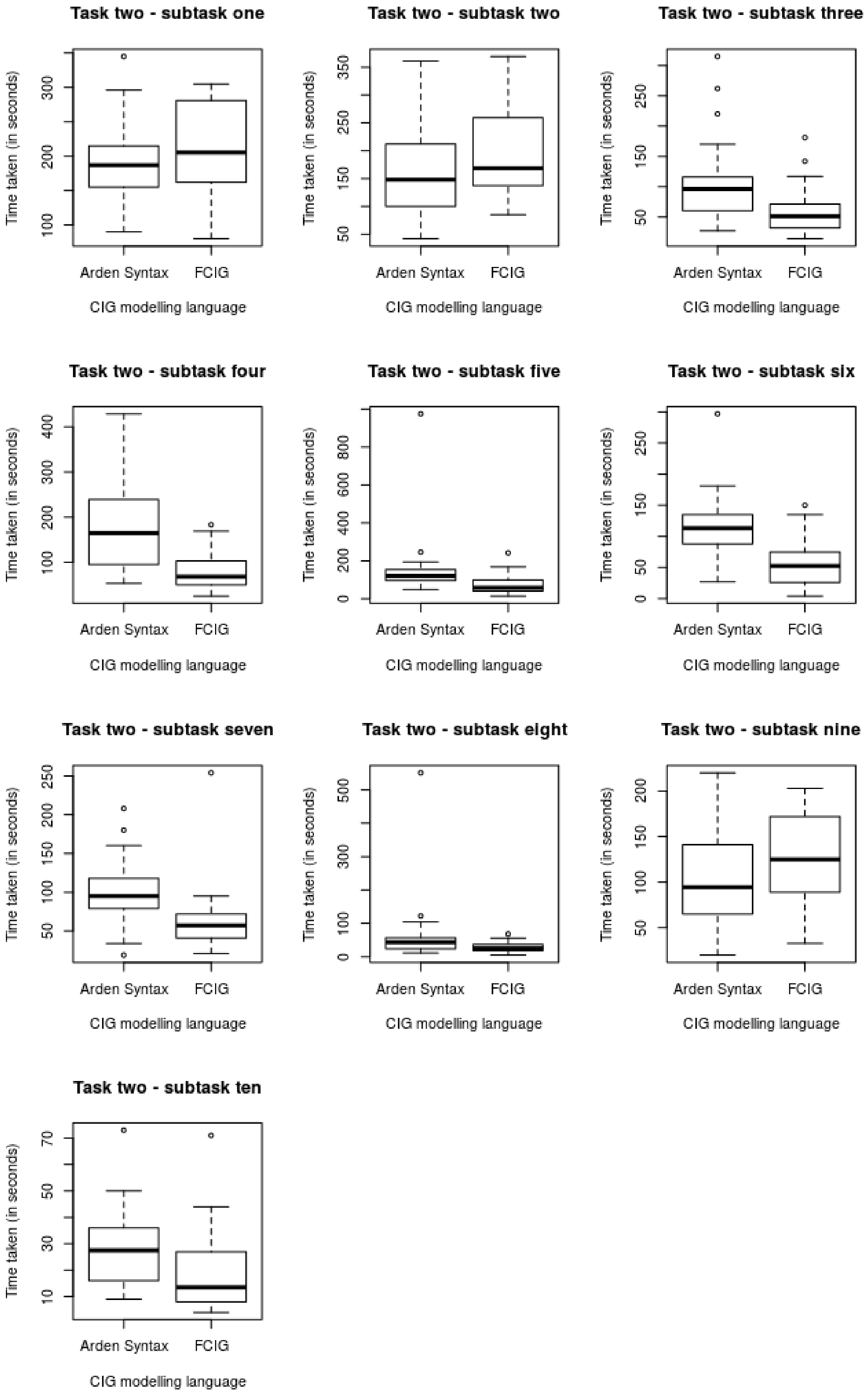


FIGURE 7.9: Box plots showing time taken on each of the subtasks for task two in the experiment

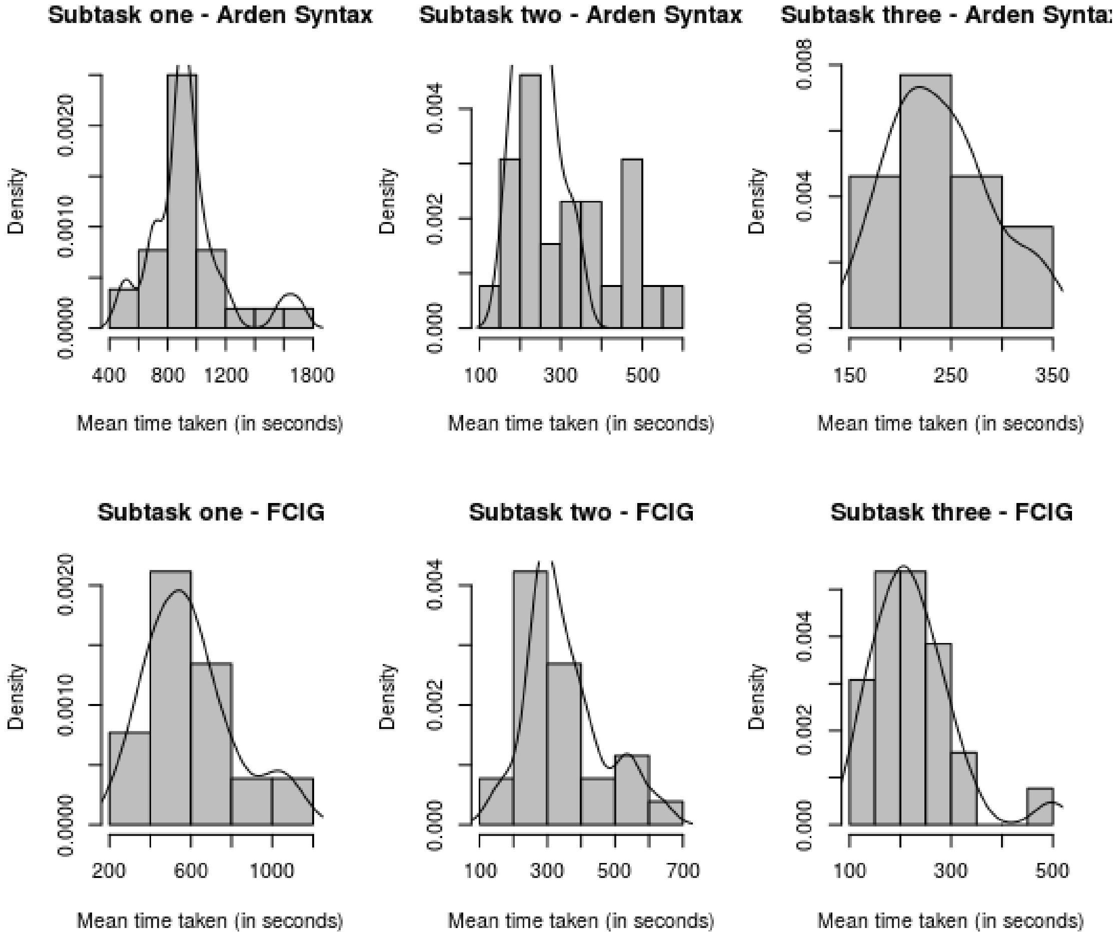


Figure 7.10: Density plots showing time taken on subtasks in task one during the experiment

Table 7.5: Shapiro-Wilk test results of time taken on subtasks

Task	Subtask	Arden Syntax		FCIG	
		W	p-value	W	p-value
One	One	0.87	0.005	0.94	0.16
	Two	0.94	0.13	0.93	0.07
	Three	0.96	0.5	0.89	0.01
Two	One	0.96	0.45	0.94	0.19
	Two	0.35	<0.001	0.9	0.02
	Three	0.85	0.001	0.29	<0.001
	Four	0.93	0.048	0.89	0.009
	Five	0.43	<0.001	0.83	0.001
	Six	0.89	0.012	0.9	0.016
	Seven	0.96	0.41	0.64	<0.001
	Eight	0.43	<0.001	0.93	0.079
	Nine	0.93	0.094	0.96	0.34
	Ten	0.92	0.036	0.37	<0.001

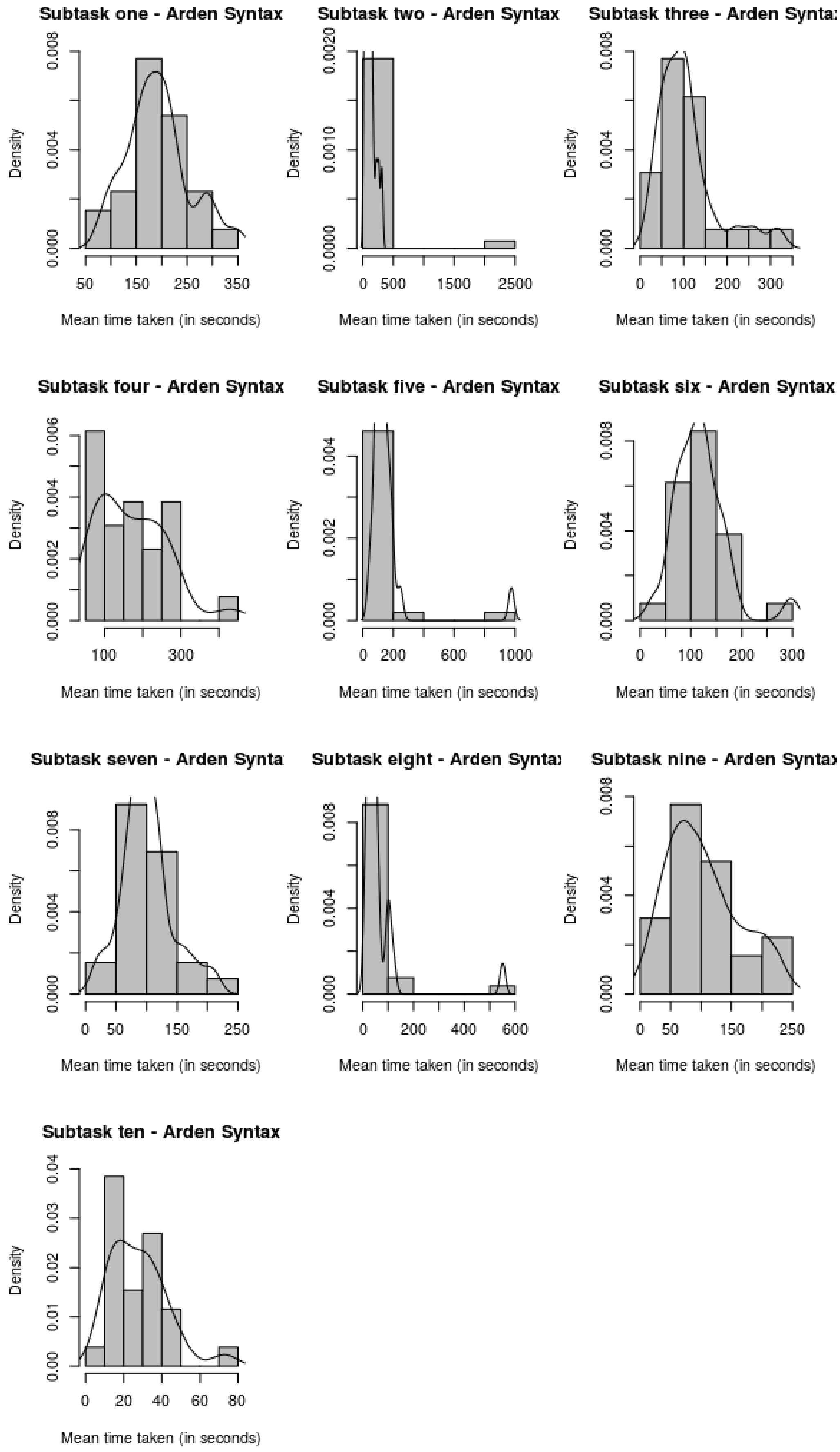


FIGURE 7.11: Density plots showing time taken on subtasks in task two with Arden Syntax in the experiment

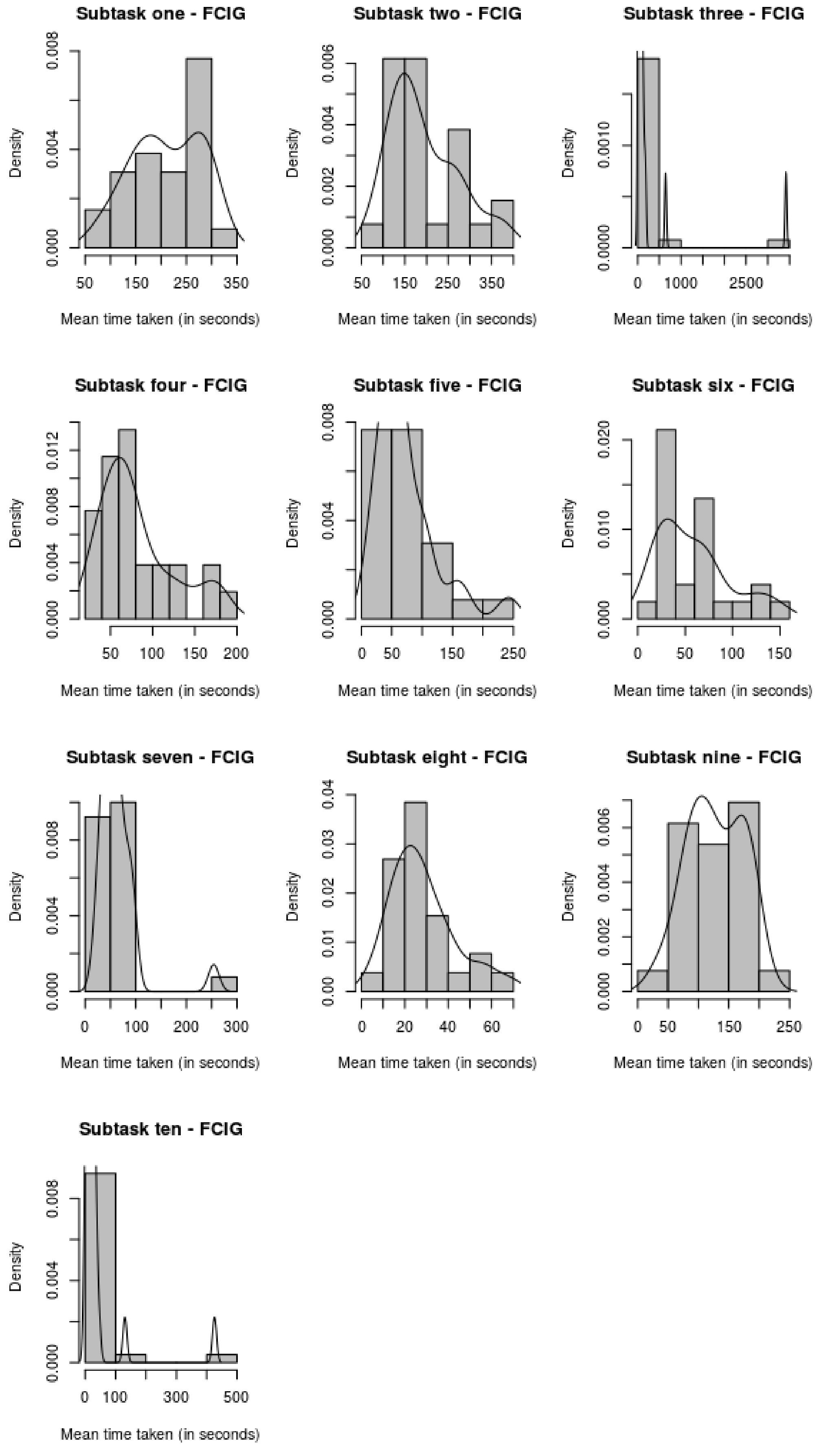


FIGURE 7.12: Density plots showing time taken on subtasks in task two with *FCIG* in the experiment

Table 7.6: Medians of subtask times

Task	**Subtask**	**Medians**	
		FCIG	**Arden Syntax**
One	One	569	904
	two	314.5	294.5
	Three	214	234.5
Two	One	205.5	187
	Two	168.5	148
	Three	54	96
	Four	68	164.5
	Five	58	120.5
	Six	52.5	113
	Seven	57	95
	Eight	25.5	43.5
	Nine	125	94.5
	Ten	15	27.5

Table 7.7: Statistical test results on subtask times with Bonferonni correction

Task	**Subtask**	**W**	**Z**	**p-value**	**r**
One	One	23.5	3.86	<0.001	0.54
	Two	231.5	1.4	0.48	0.2
	Three	110.5	1.65	0.3	0.23
Two	One	232.5	1.45	1	0.2
	Two	213.5	0.97	1	0.13
	Three	93	2.1	0.35	0.29
	Four	21	3.92	<0.001	0.54
	Five	25	3.82	<0.001	0.53
	Six	6	4.31	<0.001	0.6
	Seven	51	3.16	0.009	0.44
	Eight	62.5	2.87	0.03	0.4
	Nine	271.5	2.44	0.132	0.34
	Ten	112	1.61	1	0.22

that a particular kind of CIG modelling language does have an effect on success rate of particular types of CIG modelling functions. Specifically, the results suggest that tasks that require manipulation of fine-grained clinical practice guideline components are carried out more accurately using *FCIG* in comparison with Arden Syntax. On the other hand, the statistical test results did not provide enough evidence to reject the fourth null hypothesis on subtasks two and three of task one; and subtasks one, two, three, nine and ten of task two. These were the subtasks that required creation of composite clinical practice guideline elements such as Conditions. A closer inspection of these subtasks revealed that these subtasks were the ones that allowed participants to copy and paste clinical guideline patterns from the tutorials that were provided during the experiment. Therefore, there was not enough evidence to reject the fourth null hypothesis on subtasks that required the creation of composite elements.

7.3.3 Success rate

Recall the hypothesis for success rate:

H_0: There is no difference in success rates between *FCIG* modellers and Arden Syntax modellers. This was the null hypothesis.

H_1: The success rate of Arden Syntax modellers is more than the success rate of *FCIG* modellers. This was the alternative hypothesis.

I began to evaluate the hypothesis for success rate. All tasks were recorded as completed during the experiment on *FCIG* modellers and Arden Syntax modellers. Therefore, there was not enough evidence to warrant that the null hypothesis be rejected. We can not reject that there is no difference between the success rate of tasks completed with *FCIG* and those tasks completed with Arden Syntax. Although all participants completed all tasks for both experimental conditions, more participants completed their tasks in less time using *FCIG*.

7.3.4 Number of errors and error rates

Firstly, recall the hypothesis for error rates:

H_0: There is no difference in the error rate between *FCIG* modellers and Arden Syntax modellers. This was the first null hypothesis.

H_1: The error rate of the *FCIG* modellers is less than the error rate made by Arden Syntax modellers. This was the first alternative hypothesis.

Secondly, recall the hypothesis for number of errors on tasks:

H_0: There is no difference in the number of errors made on tasks between *FCIG* modellers and Arden Syntax modellers. This was the second null hypothesis.

H_1: The number of errors made on tasks in *FCIG* modellers is less than the number of errors made in Arden Syntax modellers. This was the second alternative hypothesis.

Thirdly, recall the hypothesis for number of errors on subtasks:

H_0: There is no difference in the number of errors made on subtasks between *FCIG* modellers and Arden Syntax modellers. This was the third null hypothesis.

$\mathbf{H}_1$: The number of errors made on subtasks between experimental *Condition A* is significantly higher than the number of errors on subtasks in *FCIG* modellers. This was the third alternative hypothesis.

I began by comparing error rates between *FCIG* modellers and Arden Syntax modellers. I used the McNemar's Chi-squared test with continuity correction to test for statistically significant differences in error rates between taswhether theks carried out with *FCIG* and those that were carried out with Arden Syntax. Furthermore, I applied the Bonferonni correction to account for multiple testing of the hypothesis. The McNemar's Chi-squared test with continuity correction revealed that the error rate on task one did not significantly differ by CIG modelling language (X^2(1, N = 26) = 4.27, p = 0.078, $\phi = 0.35$, the odds ratio is 0.83). On the other hand, the McNemar's Chi-squared test with continuity correction revealed that the error rate on task two significantly differed by CIG modelling language (X^2(1, N = 26) = 8.1, p = 0.009, $\phi = 0.4$, the odds ratio is ∞). There was significant evidence that the first null hypothesis could be rejected on error rates of task two. A closer inspection of both task one and task two revealed that the error rate was significantly less when participants modelled fine-grained clinical practice guideline components using *FCIG* during the experiment. This allowed the first alternative hypothesis to be accepted on tasks that required some manipulation of fine-grained computer-interpretable guideline components. The results suggest that there is effect of CIG modelling languages on error rate. Specifically, the results suggest that CIG modellers encounter a smaller error rate when using *FCIG* in comparison with Arden Syntax on tasks that require updates to fine-grained computer-interpretable guideline components.

I continued with the analysis in order to test the second hypothesis. I began by carrying out some exploratory analysis of the data on number of errors from the two tasks of the experiment. I created box plots showing the number of errors on tasks and subsequently checked and validated all odd data points. Figure 7.13 shows the box plots for both task one and two that show the distribution of total number of errors on tasks. I further created a set of density plots that allowed me to visualise the data on number of errors. Figure 7.14 shows density plots for total number of errors on task one and task two when completed with Arden Syntax and *FCIG*. The initial analysis showed that the samples of number of errors on ttasks might not have come from normally distributed populations.

I used the Shapiro-Wilk test to test the samples for normality. The Shapiro-Wilk test results had p-values that were below 0.05 ($p < 0.05$) on the samples for the experimental condition tests. Table 7.8 shows the details of the Shapiro-Wilk test results. The results showed that the samples of total number of errors on tasks for both task one

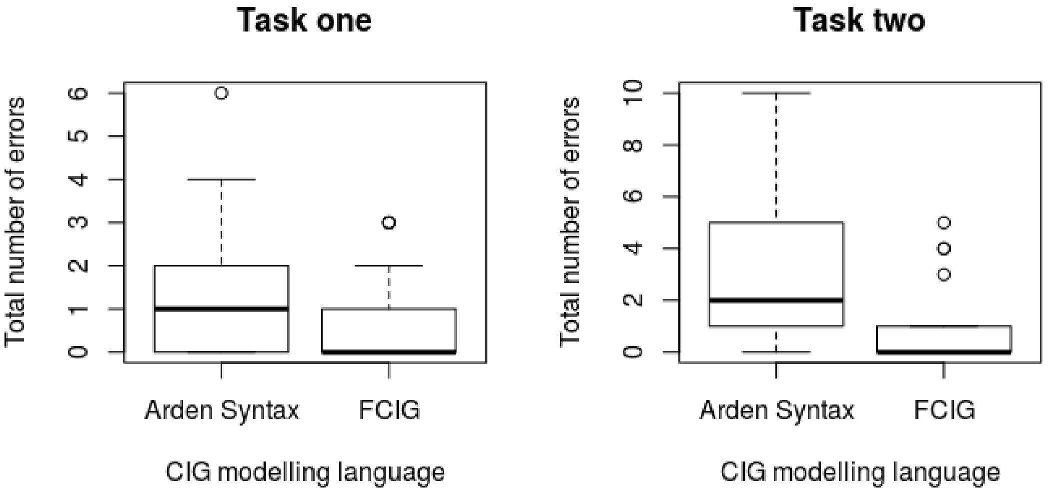


FIGURE 7.13: Box plots showing total number of errors on task in the experiment

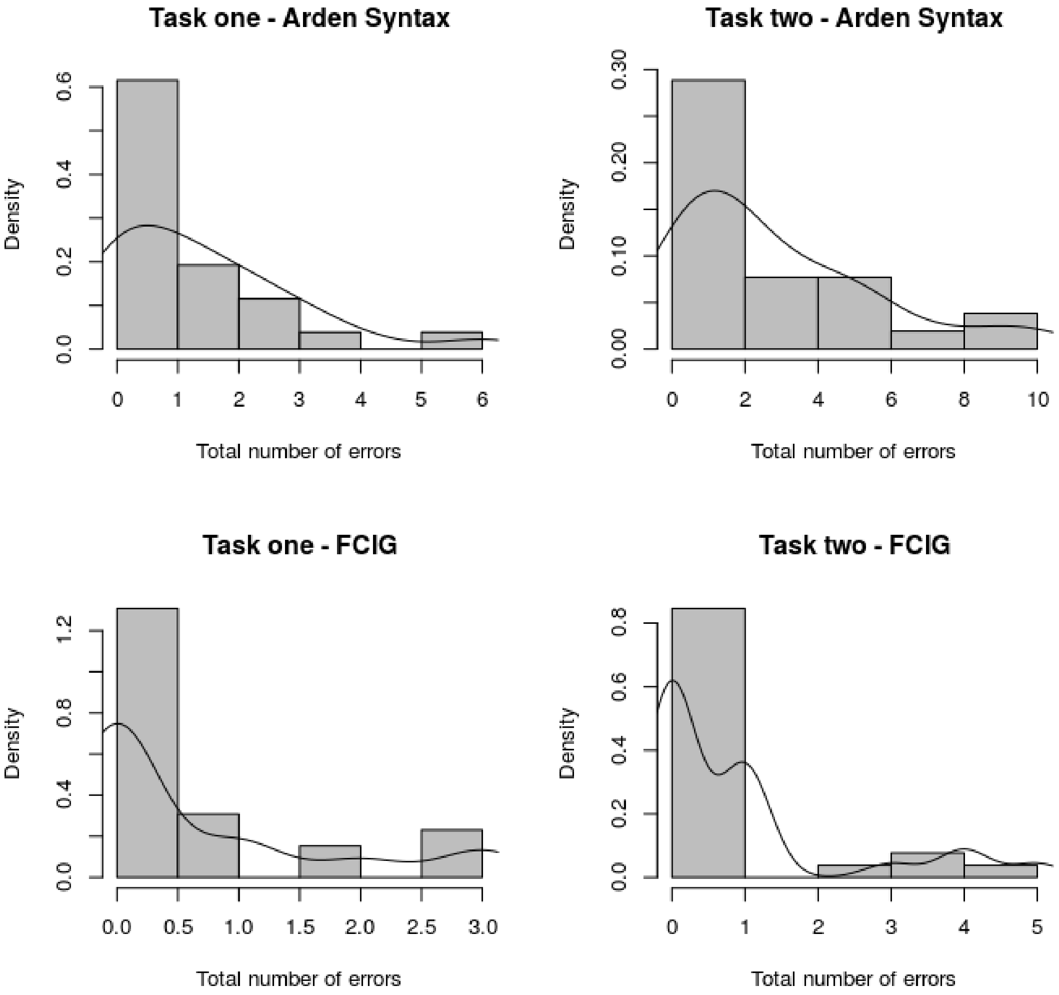


FIGURE 7.14: Density plots showing total number of errors on tasks in the experiment

Table 7.8: Shapiro-Wilk test results on total number of errors on tasks

Language	**Task**	**W**	**p-value**
Arden Syntax	Task one	0.84	<0.001
	Task two	0.85	0.002
FCIG	Task one	0.66	<0.001
	Task two	0.67	<0.001

and task two completed using both Arden Syntax and *FCIG* did not originate from normally distributed populations. Due to the fact that the number of errors on tasks did not come from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used a Wilcoxon Signed-rank test to test for statistical significance of the differences measured between the different conditions in the experiment. In addition, I also used the Bonferonni correction to account for multiple hypothesis testing.

Recall second null hypothesis on number of errors, $\mathbf{H}_0$: There is no difference in the number of errors made on tasks between *FCIG* modellers and Arden Syntax modellers. The medians of number of errors made on task one for *FCIG* and Arden Syntax were 0 and 1 respectively. A Wilcoxon Signed-rank test showed that there was no significant effect of number of errors on task one (W = 42, Z = 1.96, p = 0.11 with Bonferonni correction, r = 0.27). The medians of number of errors on task two for *FCIG* and Arden Syntax were 0 and 2 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of number of errors on task two (W = 18, Z = 3.47, $p < 0.001$ with Bonferonni correction, r = 0.48). The statistical test results did not provide enough evidence such that the second null hypothesis could not be rejected on task one. Therefore, we can say that there was not enough evidence to reject the second null hypothesis on task one. On the other hand, the statistical test results provide enough evidence such that the first null hypothesis can be rejected on task two. This means that the first alternative hypothesis can be accepted on task two. Recall that task two of the experiment comprised of computer-interpretable guideline maintenance tasks, which included maintenance of fine-grained computer-interpretable guideline elements. The results suggest that a particular kind of CIG modelling language does have an effect on the number of errors that are encountered on tasks that require manipulation of fine-grained computer-interpretable guideline components. Specifically, the results suggest that tasks that require updates to fine-grained computer-interpretable guideline components are completed with less errors using *FCIG* in comparison with Arden Syntax.

Since there was enough evidence that there were differences in the total number of errors made during the tasks between the two different conditions for the experiment, I carried out some further exploratory data analysis on the number of errors made on individual subtasks. I created box plots showing the number of errors on individual subtasks. Thereafter, I checked and validated all odd data points. Figure 7.15 and Figure 7.16 depict the box plots that show the distribution of number of errors on subtasks for task one and two respectively. I further continued with the exploratory analysis by creating density plots. Figure 7.17 shows density plots for task one when completed with both modelling languages Arden Syntax and *FCIG*. Figure 7.18 and Figure 7.19 further show the density plots for task two when completed with the CIG modelling languages Arden Syntax and *FCIG* respectively. The visual analysis of the data indicated that the samples of number of errors on subtasks might not have originated from normal distributed populations.

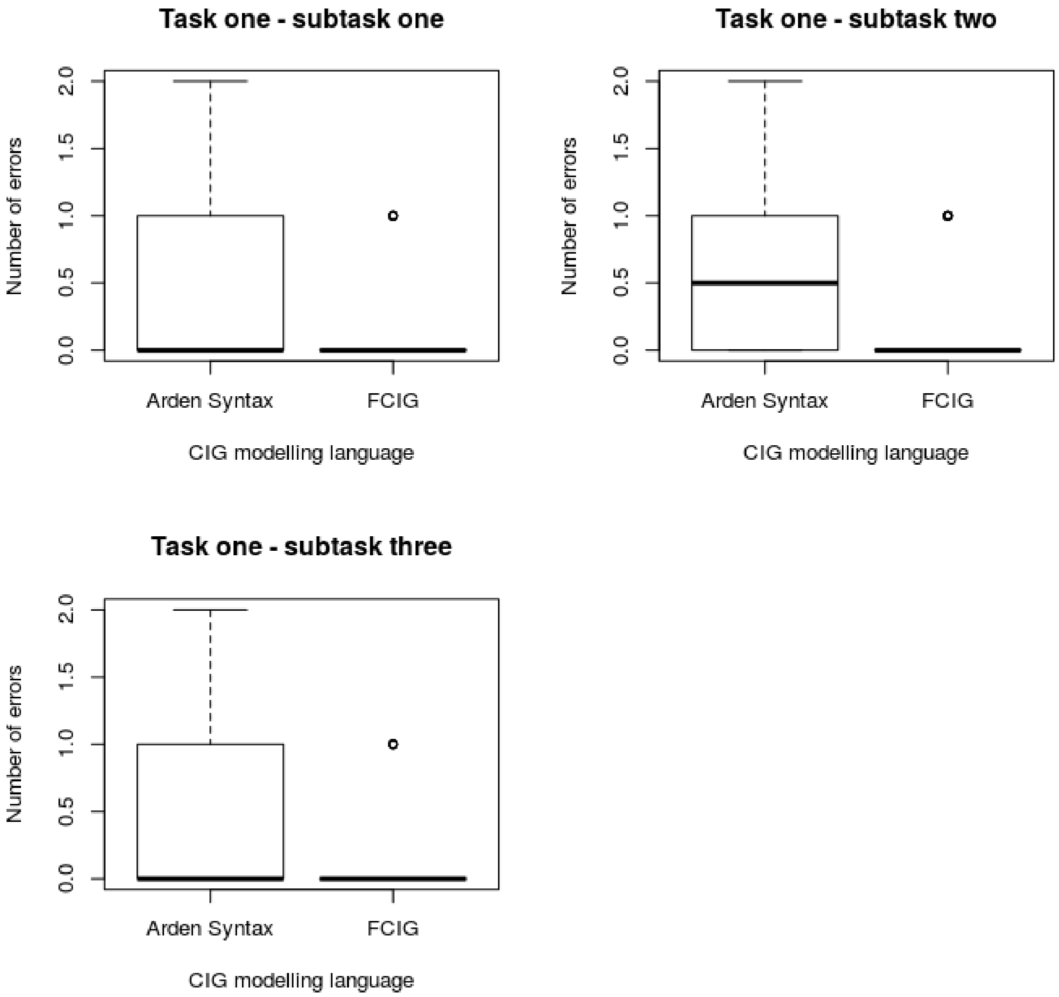


FIGURE 7.15: Box plots showing number of errors on each of the subtasks for task one in the experiment

I continued to test the samples of numbers of errors on subtasks for normality. I carried out the Shapiro-Wilk test on the number of errors that were recorded for each subtask

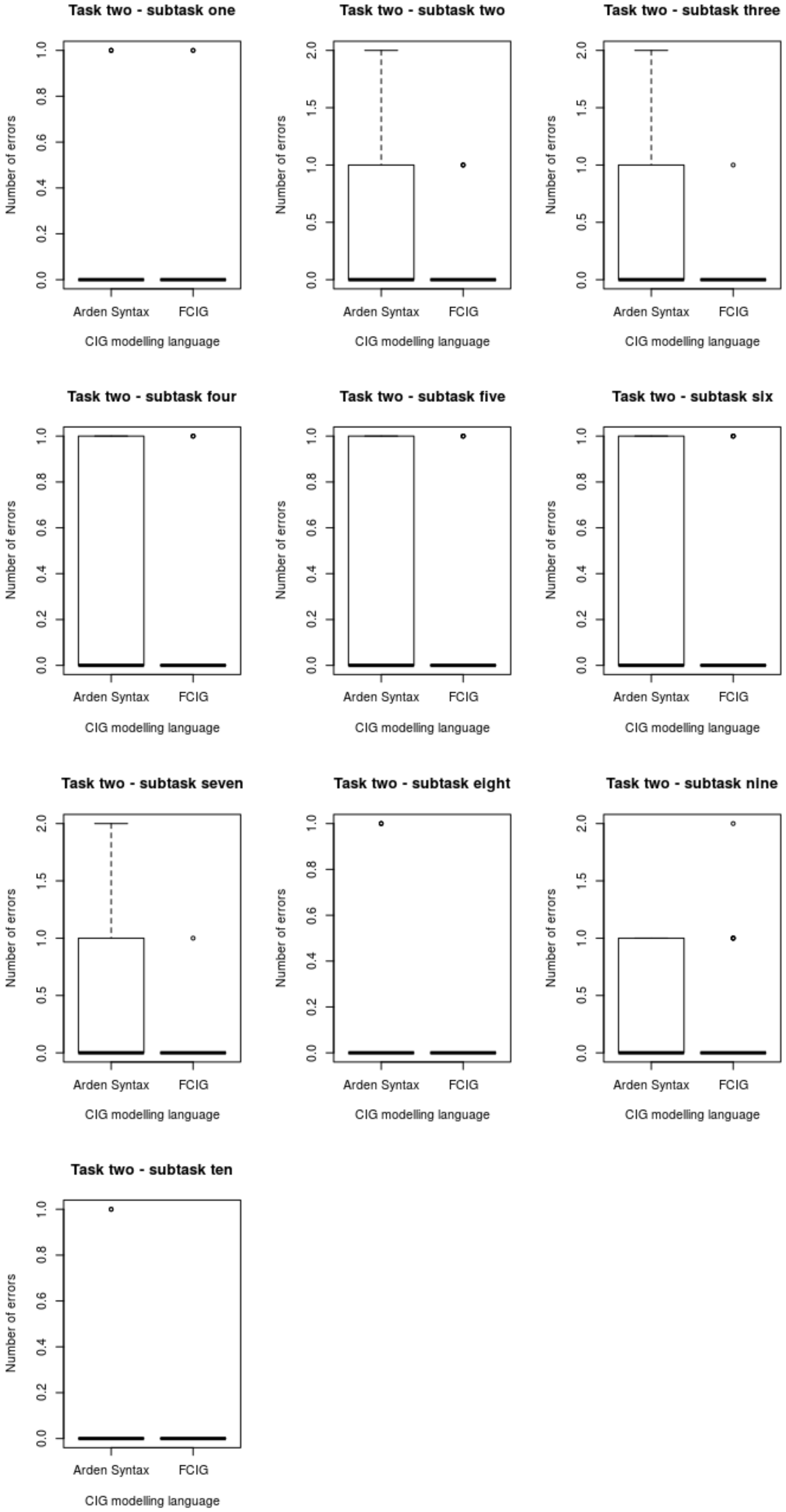


FIGURE 7.16: Box plots showing number of errors on each of the subtasks for task two in the experiment

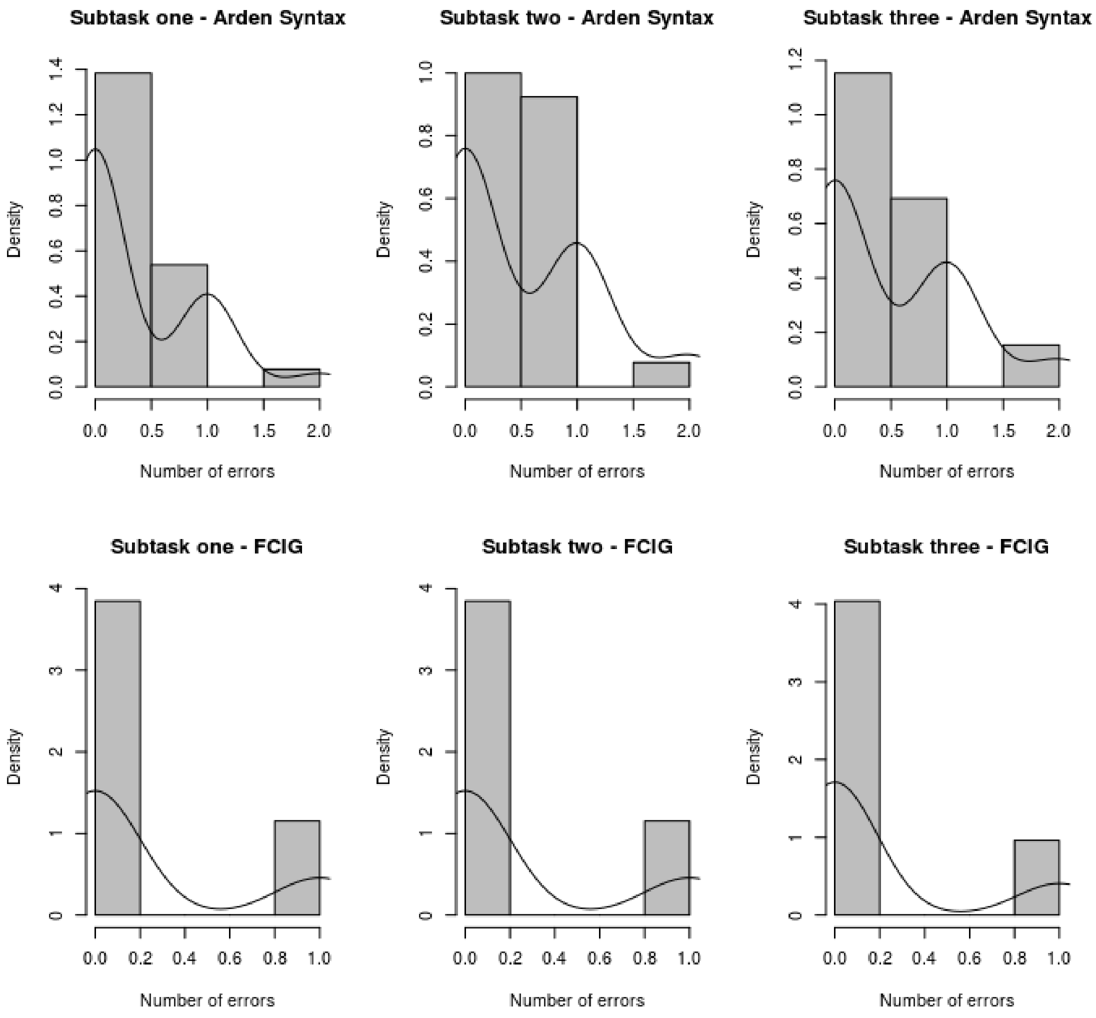


FIGURE 7.17: Density plots showing number of errors on substasks in task one during the experiment

in the experiment. The Shapiro-Wilk test results had p-values that were above 0.05 ($p > 0.05$). Table 7.9 shows the details of the Shapiro-Wilk test results. For this reason, the results show that the samples for both task one and task two carried out using both Arden Syntax and *FCIG* were not from normally distributed populations. Due to the fact that the none of the paired samples originated from normally distributed populations, I used a Wilcoxon Signed-Rank test to test for statistically significant differences in the measurements of number of errors on subtasks. I further used the Bonferonni correction to account for multiple hypothesis testing.

Recall the third null hypothesis, $\mathbf{H}_0$: There is no difference in the number of errors made on subtasks between *FCIG* modellers and Arden Syntax modellers. The medians of the time taken on each of the subtasks are presented in Table 7.11. The Wilcoxon Signed-rank test results showed that there was no significant effect of CIG modelling languages on number of errors on subtasks. Table 7.10 details the test results on subtask

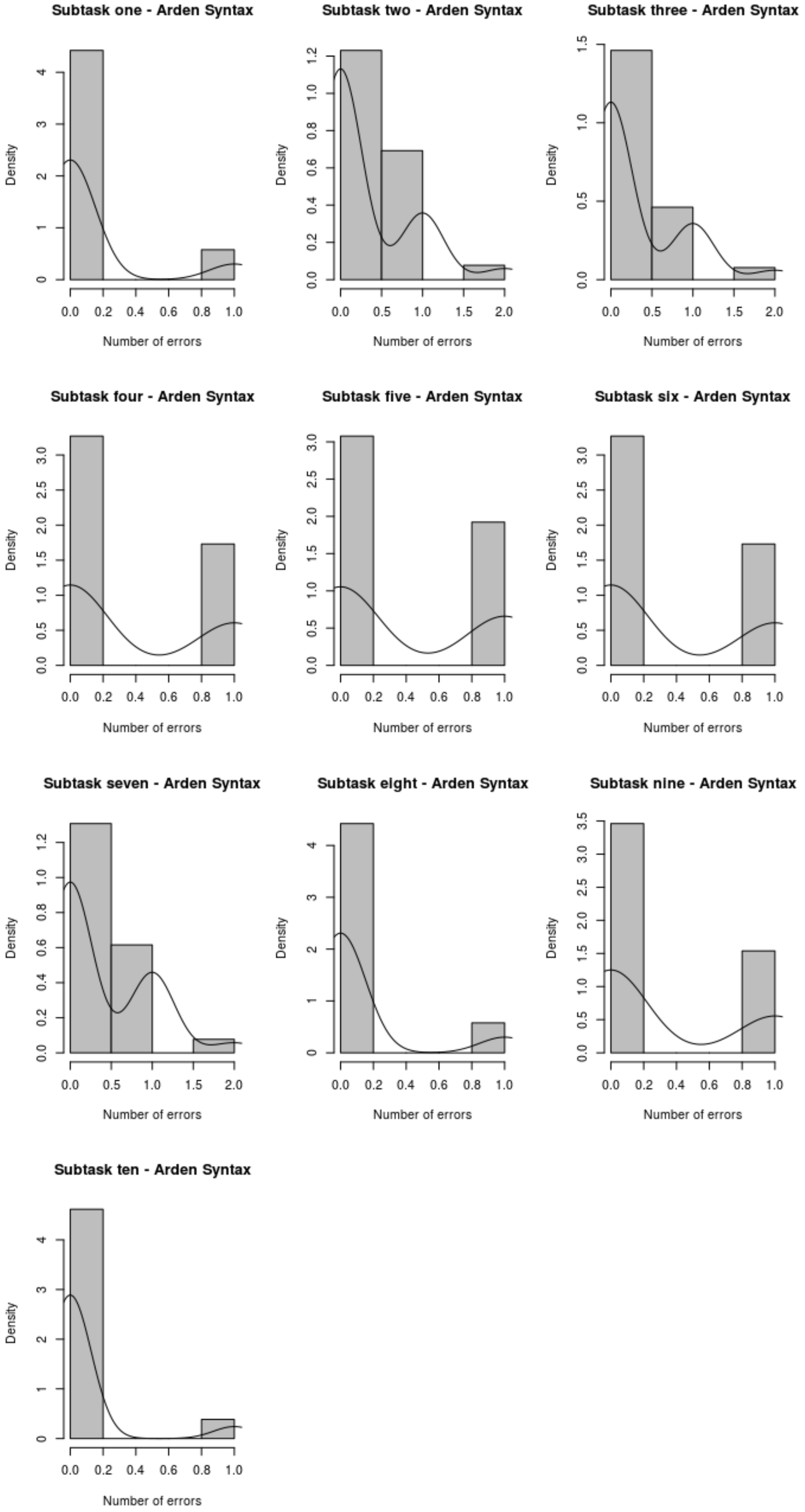


Figure 7.18: Density plots showing number of errors on subtasks in task two with Arden Syntax in the experiment

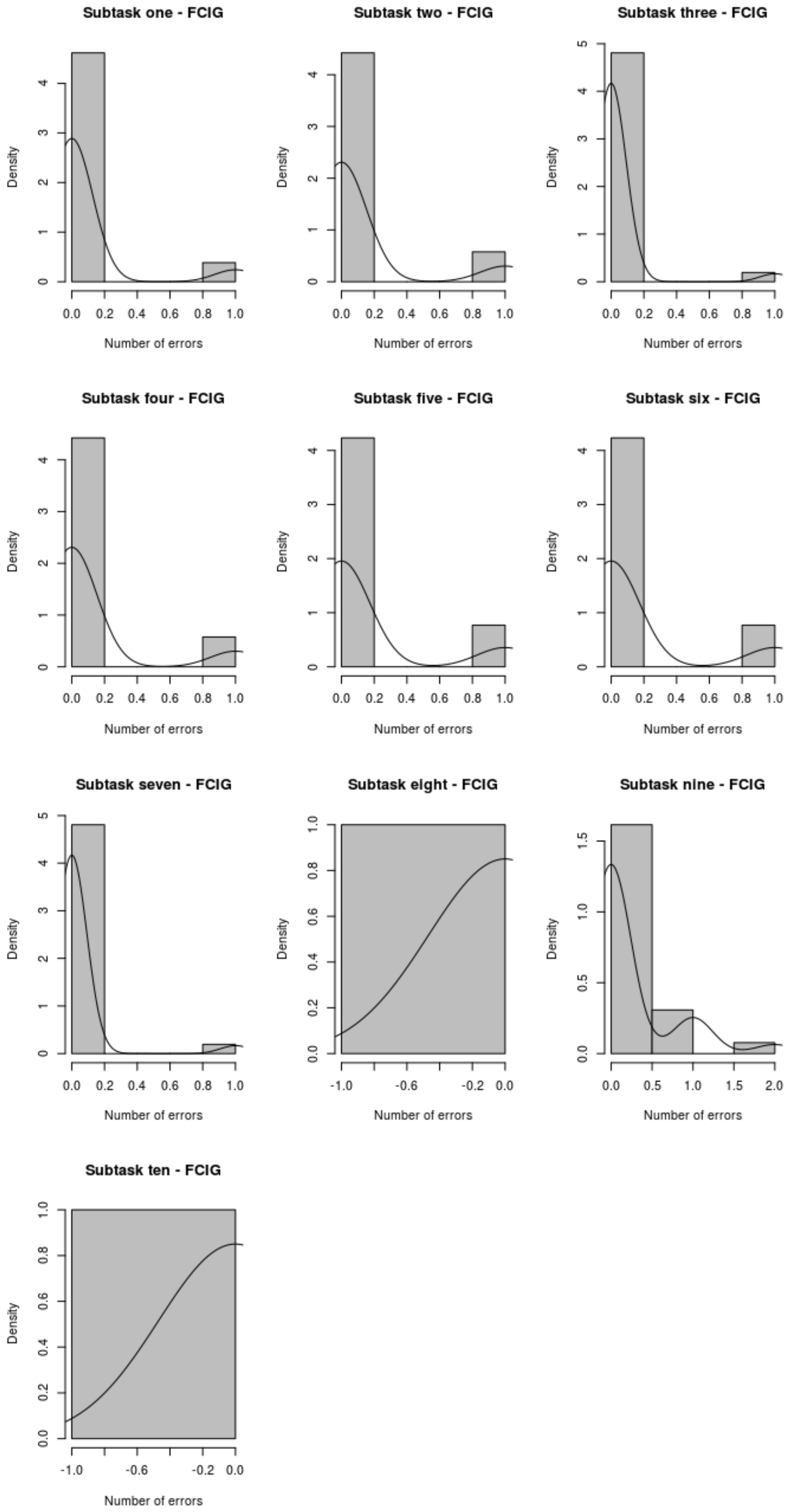


FIGURE 7.19: Density plots showing number of errors on subtasks in task two with *FCIG* in the experiment

Table 7.9: Shapiro-Wilk test results on number of errors on subtasks

Task	**Subtask**	**Arden Syntax**		**FCIG**	
		W	**p-value**	**W**	**p-value**
One	One	0.63	<0.001	0.52	0.0
	Two	0.72	<0.001	0.52	0.0
	Three	0.72	<0.001	0.48	<0.001
Two	One	0.38	<0.001	0.3	<0.001
	Two	0.68	<0.001	0.38	<0.001
	Three	0.6	<0.001	0.2	<0.001
	Four	0.6	<0.001	0.38	<0.001
	Five	0.62	<0.001	0.44	<0.001
	Six	0.6	<0.001	0.44	<0.001
	Seven	0.66	<0.001	0.2	<0.001
	Eight	0.38	<0.001		all values identical
	Nine	0.58	<0.001	0.51	<0.001
	Ten	0.3	<0.001		all values identical

Table 7.10: Statistical test results on subtask errors with Bonferonni correction

Task	**Subtask**	**W**	**Z**	**p-value**	**r**
One	One	12	0.75	1	0.1
	Two	11	2.13	0.17	0.3
	Three	16.5	1.81	0.29	0.25
Two	One	6	0.45	1	0.06
	Two	11	2.13	0.56	0.3
	Three	4	2.13	0.63	0.3
	Four	19.5	1.73	1	0.24
	Five	11	1.9	1	0.26
	Six	10	1.67	1	0.23
	Seven	5	2.54	0.2	0.35
	Eight	0	1.73	1	0.24
	Nine	13.5	0.71	1	0.098
	Ten	0	1.4	1	0.2

times. The statistical test results did not provide sufficient evidence that could warrant the rejection of the third null hypothesis.

7.3.5 Efficiency

Recall the hypothesis for efficiency:

H$_0$: There is no difference in the task success rate to task time ratio between *FCIG* modellers and Arden Syntax modellers. This is the null hypothesis.

Table 7.11: Medians of number of errors on subtask

Task	**Subtask**	**Medians**	
		FCIG	**Arden Syntax**
One	One	0	0
	two	0	0.5
	Three	0	0
Two	One	0	0
	Two	0	0
	Three	0	0
	Four	0	0
	Five	0	0
	Six	0	0
	Seven	0	0
	Eight	0	0
	Nine	0	0
	Ten	0	0

H$_1$: The task success rate to task time ratio of the *FCIG* modellers is more than the task success rate to task time ratio of the Arden Syntax modellers. This is the alternative hypothesis.

I started by carrying out some exploratory analysis of the data on efficiency from the experiment. I created box plots showing the efficiency estimates calculated as success rate to time on task ratios. I subsequently checked and validated all odd data points indicated on the box plots. Figure 7.20 shows the box plots for both task one and two that show the distribution of efficiency estimates on tasks. I continued with the visual analysis by creating density plots. Figure 7.21 shows density plots for task one and task two when completed with both modelling languages Arden Syntax and *FCIG*. The initial analysis showed that the samples of efficiency estimates on tasks might have originated from normally distributed populations.

I continued to test the samples of efficiency estimates for normality. I used the Shapiro-Wilk test to test the estimates. The Shapiro-Wilk test results had p-value higher than 0.05 ($p > 0.05$). The test results meant that samples of efficiency estimates for both task one and task two using both experimental conditions did not originate from normally distributed populations. Table 7.12 shows the details of the Shapiro-Wilk test results. Due to the fact that the efficiency estimates originated from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used a paired t-test to test for statistical significance of the differences measured in the different conditions in the experiment. I further used the Bonferonni correction to account for multiple hypothesis testing.

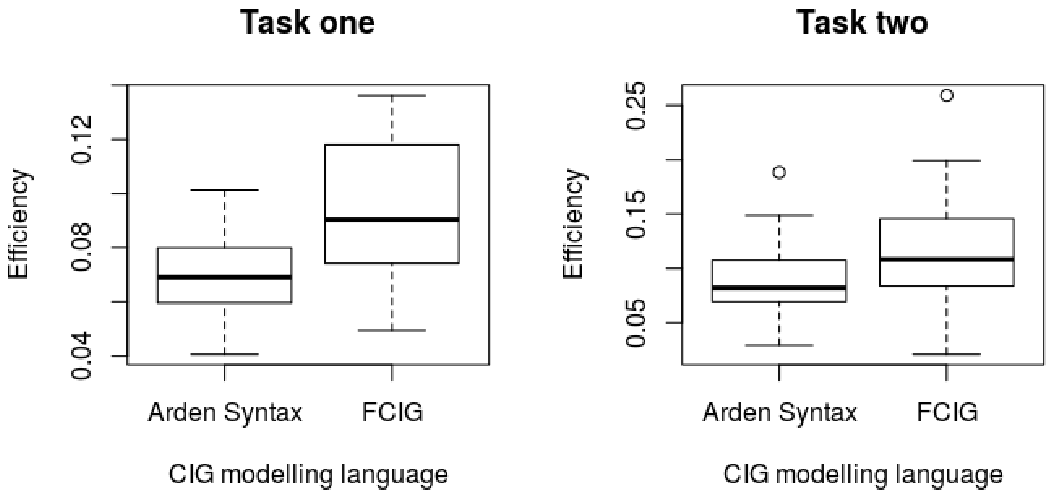


Figure 7.20: Box plots showing efficiency estimates in the experiment

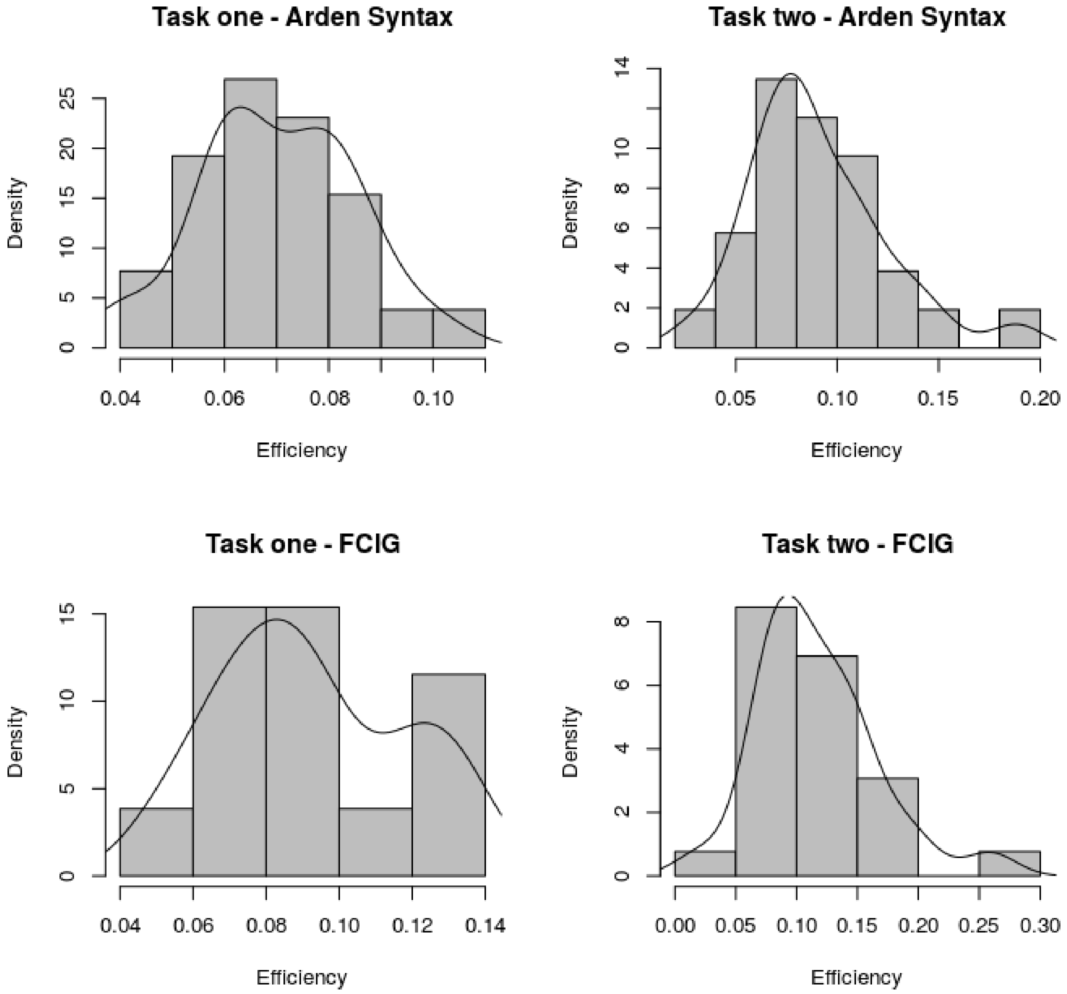


Figure 7.21: Density plots showing efficiency on tasks in the experiment

TABLE 7.12: Shapiro-Wilk test results on efficiency on tasks

Language	**Task**	**W**	**p-value**
Arden Syntax	Task one	0.98	0.78
	Task two	0.94	0.15
FCIG	Task one	0.95	0.24
	Task two	0.94	0.15

Recall the null hypothesis, $\mathbf{H}_0$: There is no difference in the task success rate to task time ratio between *FCIG* modellers and Arden Syntax modellers. The paired t-test results showed that there was a significant effect for CIG modelling languages on efficiency for task one (t = 4.74, $p < 0.001$, Cohen's d = 0.93) with *FCIG* outperforming Arden Syntax. Furthermore, the paired t-test results also showed that there was a significant effect for CIG modelling languages on efficiency for task two (t = 2.99, $p = 0.006$, Cohen's d = 0.59) with *FCIG* outperforming Arden Syntax. These results meant that the null hypothesis could be rejected such that the alternative hypothesis was accepted. The results suggest that a particular kind of CIG modelling language does have an effect on efficiency. Specifically, the results suggest that tasks are carried out more efficiently using *FCIG* in comparison with Arden Syntax.

7.3.6 Lines of code

Firstly, recall the hypothesis for lines of code (LOC):

$\mathbf{H}_0$**:** There is no difference in LOC between *FCIG* modellers and Arden Syntax modellers. This was the first null hypothesis.

$\mathbf{H}_1$**:** The LOC in *FCIG* modellers are less than the LOC of the Arden Syntax modellers. This was the first alternative hypothesis.

Secondly, recall the hypothesis for ELOC:

$\mathbf{H}_0$**:** There is no difference in ELOC between *FCIG* modellers and Arden Syntax modellers. This was the second null hypothesis.

$\mathbf{H}_1$**:** The ELOC in *FCIG* modellers are less than the ELOC of the Arden Syntax modellers. This was the second alternative hypothesis.

I started by carrying out some exploratory analysis of the data on LOC from the experiment. I created box plots showing the LOC from each task. I subsequently checked and validated all odd data points indicated on the box plots. Figure 7.22 shows the box plots for both task one and two that show the distribution of LOC from tasks. I continued with the exploratory analysis of the samples by creating some density plots. Figure 7.23

shows density plots for task one and task two when completed with both modelling languages Arden Syntax and *FCIG*. The initial analysis showed that the samples of LOC for tasks might not have come from normally distributed populations.

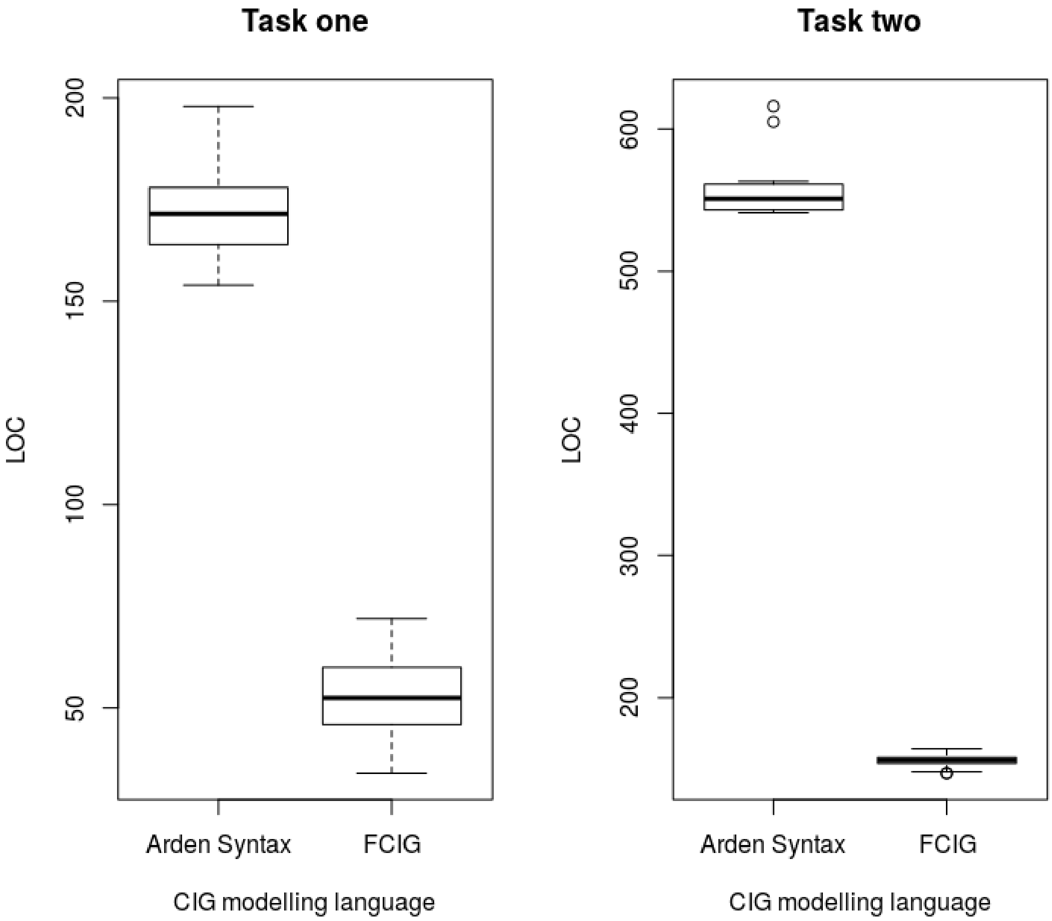


Figure 7.22: Box plots showing LOC in the experiment

I continued to test the samples for normality by using the Shapiro-Wilk test on the samples of lines of code measurements. The Shapiro-Wilk test results had p-values of less than 0.05 ($p < 0.05$) for the estimates from *FCIG* modellers and Arden Syntax modellers. Table 7.13 shows the details of the Shapiro-Wilk test results. The test results showed that not all the LOC samples for both task one and task two were from normally distributed populations. Due to the LOC measurements of tasks not coming from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used the Wilcoxon Signed-Rank test to test for statistically significant differences in the LOC measurements. I also used the Bonferonni correction to account for multiple hypothesis testing.

Recall the first null hypothesis, $\mathbf{H}_0$: There is no difference in LOC between *FCIG* modellers and Arden Syntax modellers. The medians of LOC on task one for *FCIG* and

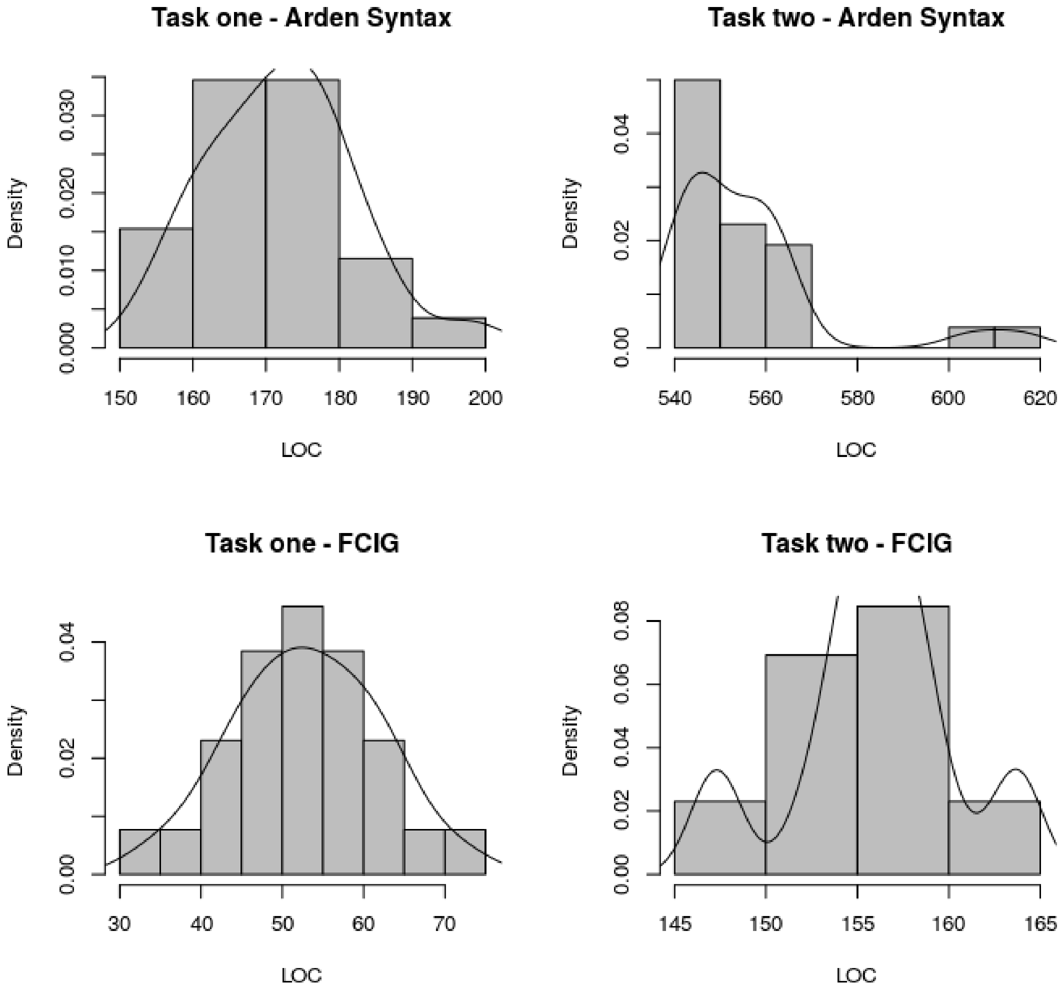


Figure 7.23: Density plots showing LOC on tasks in the experiment

Table 7.13: Shapiro-Wilk test results on LOC on tasks

Language	**Task**	**W**	**p-value**
Arden Syntax	Task one	0.98	0.8
	Task two	0.7	<0.001
FCIG	Task one	1	1
	Task two	0.94	0.13

TABLE 7.14: Shapiro-Wilk test results on ELOC on tasks

Language	**Task**	**W**	**p-value**
Arden Syntax	Task one	0.91	0.028
	Task two	0.75	<0.001
FCIG	Task one	0.76	<0.001
	Task two	0.72	<0.001

Arden Syntax were 52.5 and 171.5 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of CIG modelling language on LOC for task one (W = 0, Z = 4.46, $p < 0.001$ with Bonferonni correction, r = 0.62). The medians of LOC on task two for *FCIG* and Arden Syntax were 156 and 551 respectively. A Wilcoxon Signed-rank test also showed that there was a significant effect of CIG modelling language on LOC for task two (W = 0, Z = 4.45, $p < 0.001$ with Bonferonni correction, r = 0.62). The statistical test results provide enough evidence such that the first null hypothesis can be rejected. This means that the first alternative hypothesis can be accepted. The results suggest that a particular kind of CIG modelling language does have an effect on LOC. Specifically, the results suggest that tasks are completed in less lines of code using *FCIG* in comparison with Arden Syntax.

I continued to carry out some exploratory analysis of the data on effective lines of code (ELOC). I started by creating box plots showing the distribution of ELOC from the measurements of each task. Thereafter, I checked and validated all odd data points indicated on the box plots. Figure 7.24 shows the box plots for both task one and two that show the distribution of ELOC from tasks. I continued with the exploratory analysis of the samples by creating some density plots. Figure 7.25 shows density plots for task one and task two when completed using both modelling languages, Arden Syntax and *FCIG*. The initial analysis showed that the samples of ELOC for tasks might not have originated from normally distributed populations.

I continued to test the samples for normality by using the Shapiro-Wilk test on the samples of effective lines of code measurements. The Shapiro-Wilk test results had p-values of less than 0.05 ($p < 0.05$) for the estimates from *FCIG* modellers and Arden Syntax modellers. Table 7.14 shows the details of the Shapiro-Wilk test results. The test results showed that not all the ELOC samples for both task one and task two originated from normally distributed populations. Due to the ELOC measurements of tasks not coming from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used the Wilcoxon Signed-rank test to test for statistically significant differences in the ELOC measurements. I also used the Bonferonni correction to account for multiple hypothesis testing.

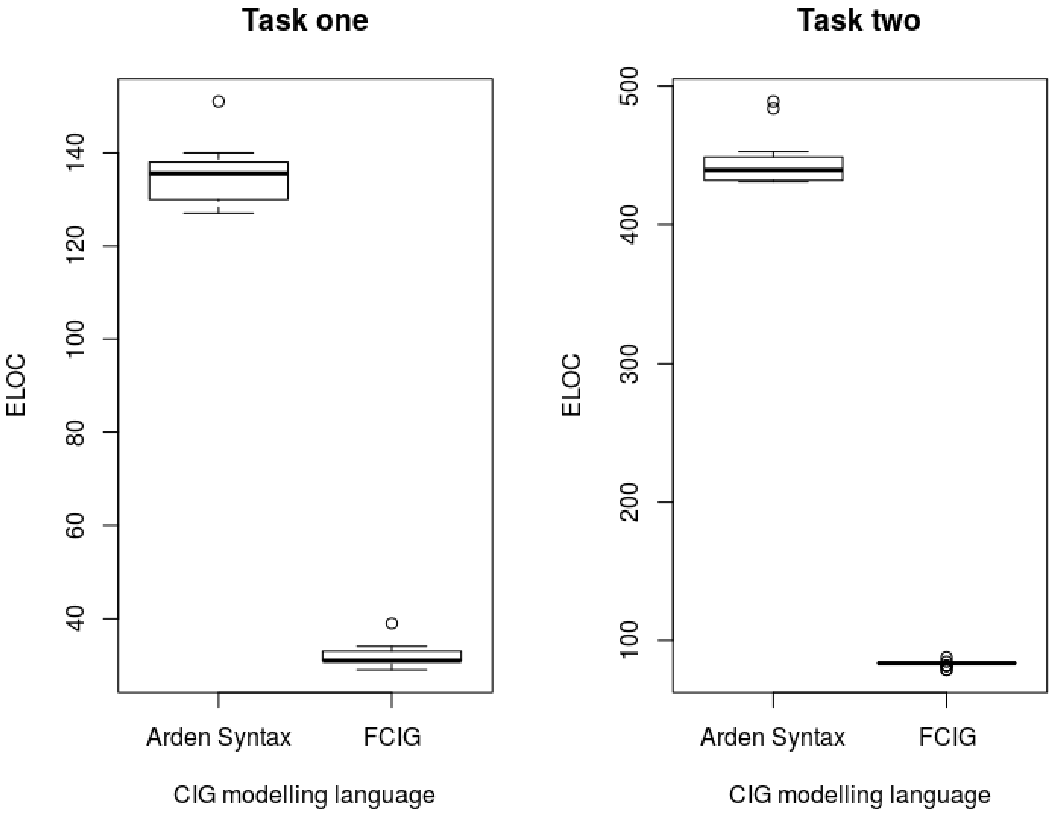


Figure 7.24: Box plots showing ELOC in the experiment

Recall the second null hypothesis, $\mathbf{H}_0$: There is no difference in ELOC between *FCIG* modellers and Arden Syntax modellers. The medians of ELOC from task one for *FCIG* and Arden Syntax were 31 and 135.5 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of CIG modelling language on ELOC for task one (W = 0, Z = 4.46, $p < 0.001$ with Bonferonni correction, r = 0.62). The medians of ELOC from task two for *FCIG* and Arden Syntax were 84 and 439.5 respectively. A Wilcoxon Signed-rank test also showed that there was a significant effect of CIG modelling language on ELOC for task two (W = 0, Z = 4.46, $p < 0.001$ with Bonferonni correction, r = 0.62). The statistical test results provided enough evidence such that the second null hypothesis could be rejected. This means that the second alternative hypothesis could be accepted. The results suggest that a particular kind of CIG modelling language does have an effect on ELOC. Specifically, the results suggest that tasks are completed in less effective lines of code using *FCIG* in comparison with Arden Syntax.

7.3.7 SUS scores

Recall the hypothesis for SUS scores:

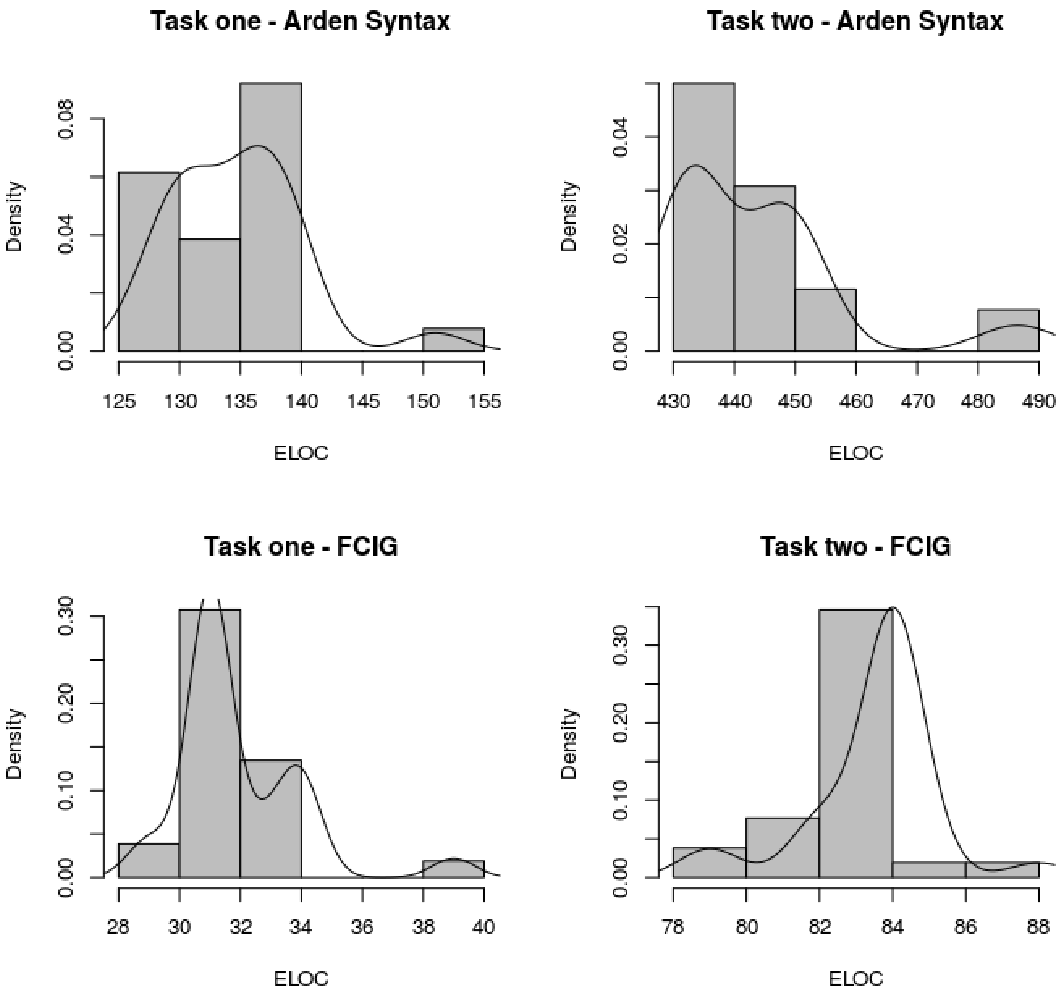


Figure 7.25: Density plots showing ELOC on tasks in the experiment

H$_0$: There is no difference in SUS scores between *FCIG* modellers and Arden Syntax modellers. This is the null hypothesis.

H$_1$: The SUS scores from *FCIG* modellers are higher than the SUS scores from the Arden Syntax modellers. This is the alternative hypothesis.

I started by carrying out some exploratory analysis of the data on SUS scores from the experiment. I created box plots showing the SUS scores from each task. I subsequently checked and validated all odd data points indicated on the box plots. Figure 7.26 shows the box plots for that show the distribution of SUS scores for each CIG modelling language. I continued with the exploratory analysis by creating density plots. Figure 7.27 shows the density plots for SUS scores for both Arden Syntax and *FCIG*. The initial analysis showed that the samples of efficiency estimates on tasks might have not originated from normally distributed populations.

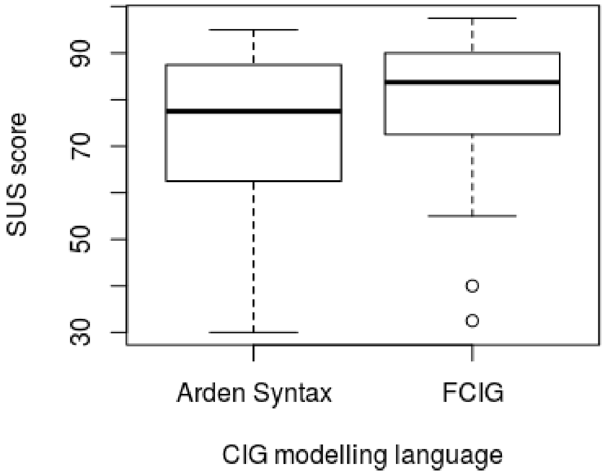


FIGURE 7.26: Box plots showing SUS scores in the experiment

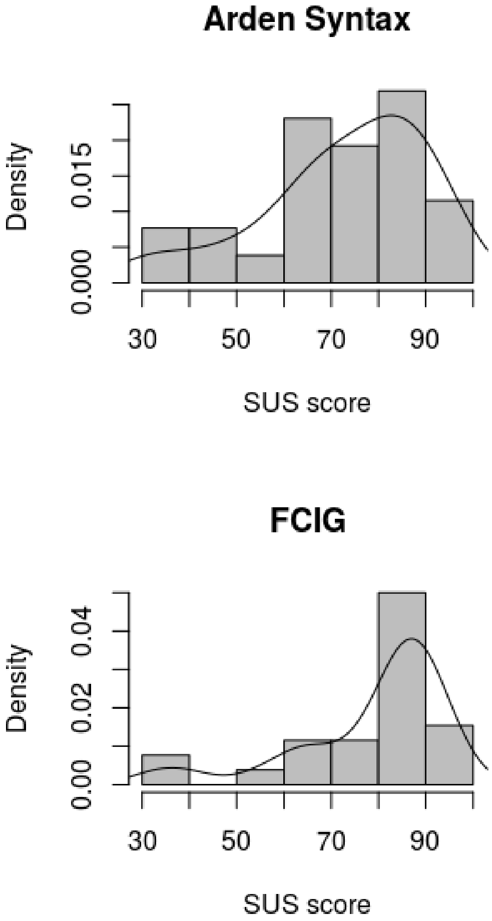


FIGURE 7.27: Density plots showing SUS scores in the experiment

TABLE 7.15: Shapiro-Wilk test results on SUS scores for the CIG modelling languages

Language	**W**	**p-value**
Arden Syntax	0.91	0.034
FCIG	0.83	<0.001

I continued to test the samples for normality using the Shapiro-Wilk test. The Shapiro-Wilk test results had p-values that were less than 0.05 ($p < 0.05$). For this reason, the results show that not the all SUS samples for both Arden Syntax and *FCIG* did not originate from normally distributed populations. Table 7.15 shows the details of the Shapiro-Wilk test results. Due to the SUS scores of tasks not coming from normally distributed populations, and the fact that I chose a repeated measures design for the experiment, I used a Wilcoxon Signed-rank test to test for statistically significant differences in SUS scores between *FCIG* and Arden Syntax.

The medians of SUS scores for *FCIG* and Arden Syntax were 83.75 and 77.5 respectively. A Wilcoxon Signed-rank test showed that there was a significant effect of CIG modelling languages on SUS score (W = 204, Z = 2.02, p = 0.042, r = 0.28). The statistical test results provide enough evidence such that the null hypothesis could be rejected. This meant that the alternative hypothesis could be accepted. The results suggest that a particular kind of CIG modelling language does have an effect on SUS scores. Specifically, the results suggest that CIG modellers perceive *FCIG* as a more usable modelling language in comparison with Arden Syntax.

7.3.8 Qualitative analysis results

The thematic analysis allowed for a systematic comparison of *FCIG* against the HL7 standard, Arden Syntax, to be carried out. A number of interesting insights into how the computer-interpretable guideline (CIG) modellers perceived the usability of both CIG modelling languages when carrying out CIG modelling tasks were uncovered. The rest of this section discusses the six themes that were derived from the thematic analysis.

The first theme was identified as 'simplicity'. This theme was concerned with the simplicity of the CIG modelling languages. A number of CIG modellers found *FCIG* to be a simpler CIG modelling language in comparison with Arden Syntax. In particular, the CIG modellers found *FCIG* as a simple, flexible and easy to understand syntax when creating and maintaining CIG models. CIG modellers labelled *FCIG* as a *"straight forward"*, *"intuitive"*, *"pleasant enough"*, *"flexible"* and an *"easy to use and modify"* CIG modelling language. Simplicity is a key characteristic of a DSL that has to be considered in order to realise a pragmatic domain-specific language (DSL) [1, 216, 230].

The second theme focused on 're-usability'. This theme was centred around the re-usability of semantic elements of a CIG. Some CIG modellers expressed that *FCIG* allowed them to reuse Condition and Action constructs of a CIG where as Arden Syntax did not. The CIG modellers indicated that they found the ability to reuse previously constructed CIG elements in *FCIG* very helpful as that feature enabled them to complete the CIG modelling and maintenance tasks faster. Software engineering concerns such as re-usability are desirable characteristics that contribute to usable DSLs [182, 209].

The third theme was identified as 'domain-appropriateness'. This theme was concerned with the overall language features that enable the formalization of a clinical practice guideline into a CIG. A number of CIG modellers indicated that *FCIG* felt more of a natural-fit when modelling CIGs as compared to the times they used Arden Syntax. The CIG modellers further indicated that all of *FCIG*'s language constructs were quite useful when modelling evolving CIGs. Domain-appropriateness of a DSL is a key characteristic that can assist in making a particular DSL to be a tool of choice in a particular domain [1, 215].

The fourth theme was identified as 'extensions'. This theme was concerned with the possibility of adding GPL-like language constructs into the core DSL vocabulary of the CIG modelling languages. A number of CIG modellers indicated that some language features such as composite conditions, negation and range selection can be beneficial to the CIG modelling process. Though other CIG modellers were of the opinion that *FCIG* requires more language constructs, a number of CIG modellers found Arden Syntax to be too verbose as it requires a lot of boilerplate code. Inclusion of GPL-like language constructs as language modelling requirements for *FCIG* would have to be carefully evaluated and balanced so as to not affect language simplicity and compactness negatively. Simplicity and compactness are key factors that should be considered in the design of a DSL in order to increase the likelihood of DSL adoption [1, 230].

The fifth theme was named 'adequacy'. This particular theme was concerned with the adequacy of the languages in modelling evolving CIGs. Quite a number of CIG modellers felt that *FCIG* was adequate for modelling evolving CIGs and that the language did not need new features to support CIG modellers in modelling evolving CIGs. Semantic correctness and completeness of models that are created using a particular DSL are key to determining the DSL's success [1, 310].

The sixth theme was identified as 'smart-editing support'. This theme focused on availability of smart-editing features in the CIG modelling tools when modelling CIGs using the two languages. The CIG modellers found that smart-editing support for features such as auto-complete, suggestion and outline for both *FCIG* and Arden Syntax contributed to simplifying the evolving CIG modelling process when using both languages.

But some CIG modellers found the outline feature to be misleading when using Arden Syntax. For instance, one participant had the following to say about Arden Syntax: *"I kept losing track of where to edit"*. Modelling tools for a DSL are key to the DSL's acceptance in its modelling community [153, 221, 222].

On the whole, CIG modellers preferred *FCIG* when modelling CIGs. This was an expected result noting that *FCIG*'s language model is based on a CIG conceptualisation that closely matches how clinical practice guidelines are presented in low- and middle-income countries. Recall the discussion on clinical practice guidelines in Chapter 2. Clinical practice guidelines for many conditions in low- and middle-income countries are designed for use by different levels of health workers which include task-shifted health workers with limited training and at times with limited access to diagnostic testing tools [54]. In addition, these CPGs are often revised as new evidence emerge on how to manage the various conditions targeted in the CPGs [57–59]. *FCIG*'s language model explicitly specifies simplified clinical practice guideline representation primitives that also include the fine-grained components that are typically affected by CPG changes.

7.4 Chapter summary

An experiment was carried out to compare *FCIG* with the HL7 standard for modelling CPGs, Arden Syntax. The experiment was conducted with a total of 26 participants recruited from the University of Cape Town in South Africa. All the participants were students that novice CIG modellers that had no prior experience with medical computing systems. On the other hand, the novice CIG modellers were familiar with at least two general-purpose computer programming languages. During the experiment, novice CIG modellers were asked to complete CIG modelling tasks using both *FCIG* and Arden Syntax using a repeated measures experiment design.

Results from this experiment provided enough evidence that novice CIG modellers performed better at CIG encoding and maintenance tasks when using *FCIG* in comparison with Arden Syntax. The performance measurements were assessed across the following six dimensions: *i*) task time, *ii*) success rate, *iii*) errors on task, *iv*) efficiency, *v*) lines of code, *vi*) usability perception.

When considering time on task, the results suggest that CIG modellers take less time on modelling tasks when using *FCIG* in comparison with Arden Syntax. Further analysis also suggest that CIG modellers spend less time handling fine-grained CIG components when using *FCIG* in comparison with Arden Syntax. Such was not the case with success rate as the results suggest that that there is no effect of CIG modelling languages on task completion rates between those tasks that are attempted with *FCIG* in comparison with those that are attempted with Arden Syntax.

Two aspects with regards to errors on task were considered. The first set of results, concerning error rate, suggest that there is effect of CIG modelling languages on error rate. To be specific, CIG modelling tasks are completed with a lower error rate when using *FCIG* in comparison with Arden Syntax. The second set of results, concerning number of errors, suggest that participants completed tasks with fewer errors when using *FCIG* in comparison with Arden Syntax on tasks that required updates to fine-grained CIG components.

When considering efficiency, the results suggest that CIG modellers completed tasks more efficiently when using *FCIG* as compared to Arden Syntax. Another related set of measures were lines of code (LOC) and effective lines of code (ELOC), whose results suggest that CIG modellers complete tasks with fewer lines of source code when using *FCIG* as compared with Arden Syntax.

The last dimension that was measured was participant perception on CIG modelling language usability. The results from this experiment provide enough evidence that the CIG modellers perceived *FCIG* as more usable when compared with Arden Syntax.

The results of the thematic analysis that was carried out on the qualitative data uncovered a number of interesting insights with regards to the perceptions the CIG modellers had on the two CIG modelling languages. CIG modellers found *FCIG* to be a natural fit with its simplicity and reusable language constructs when modelling evolving CIGs.

The chapter that follows provides a synthesis of how the empirical results address the overall research questions that were posed for this work. The following chapter further concludes this thesis.

Chapter 8

Conclusion

This work proposed a novel model-based DSL, *FCIG*, in a four-layer guideline modelling architecture that can be used for modelling evolving CIGs in low- and middle countries. In order to address this proposition, the following three research questions were set:

1. What are the CPG change requirements for modelling an evolving CIG?
2. Can a model-driven engineering approach adequately support the modelling of an evolving CIG?
3. What is the effect of modelling an evolving CIG using *FCIG* in comparison with the HL7 standard for modelling CIGs?

This chapter starts with a synthesis of how the empirical findings from this work addressed the research questions. Further to that, a discussion on the implications of the study follows on. Finally, the limitations of the study and implications for future work are discussed.

8.1 Synthesis of empirical results

This section synthesises empirical findings from this work in relation to their research questions.

8.1.1 What are the CPG change requirements for modelling an evolving CIG?

The findings for this research question indicate that fine-grained semantic elements of a CPG are affected when changes occur. The findings further revealed that there are

10 types of CPG changes that occur, which are outlined in Table 8.1. In addition, the findings showed that the CPG changes affected fine-grained semantic elements of a CPG such as *decision variables*, *variable values*, *action verbs* and *verb complements*. Furthermore, the findings indicated that existing CIG modelling languages, that are mostly experimental, lack explicit language construct specifications for the fine-grained CPG components that are affected when the changes occur. Table 8.2 shows the representation primitives in existing CIG modelling languages.

Table 8.1: Summary of CPG changes categories.

Category	**Semantic element**	**Type of change**
Decision	Decision variable	Addition of a decision variable to a guideline condition
	Variable value	Change of a decision variable value
	Decision variable	Removal of a decision variable from a guideline condition
	Decision variable	Change of a decision variable
Action	Action	Addition of a recommended action
	Action	Removal of a recommended action
	Verb complement	Change of an action verb complement
	Action	Change of a recommended action
Recommendation	Recommendation	Addition of a recommendation
	Recommendation	Removal of a recommendation

Table 8.2: Representation primitives in existing CIG modelling languages.

Structural Component	**Arden Syntax**	**GLIF**	**SAGE**	**EON**	**PROforma**	**Asbru**
Condition	Yes	Yes	Yes	Yes	Yes	Yes
Decision variable	No	No	No	No	No	No
Variable value	No	No	No	No	No	No
Action	Yes	Yes	Yes	Yes	Yes	Yes
Action verb	No	No	No	No	No	No
Verb complement	No	No	No	No	No	No

8.1.2 Can a model-driven engineering approach adequately support the modelling of an evolving CIG?

In order to investigate the answers to this research question, a four-layer architecture for modelling evolving CIGs that included *FCIG* as its DSL was created. Further to that, CPG documents were obtained from the Malawi Ministry of Health for experimentation. The resulting findings that answered this research question are synthesised through discussions of its related sub-questions that follow.

Can FCIG be used to model an evolving CIG adequately?

A representative sample of the Malawi CPGs were formalized into CIGs using *FCIG*. The findings from this work showed that *FCIG* can be used to model evolving CIGs adequately.

Can FCIG directly support the application of fine-grained CPG changes?

A representative sample of CPG changes between successive sets of CPGs were applied to existing *FCIG* encoded CIGs. The findings from this work showed that *FCIG* language constructs were available for all concepts of an evolving CIG. Furthermore, the findings showed that smart editing features for supporting all CPG change operations were available in the language-aware code editor of the novel CIG modelling environment.

Are the language constructs of FCIG perceived as usable?

Novice and experienced CIG modellers were introduced to *FCIG*'s language constructs. Both novice and experienced CIG modellers perceived *FCIG* as highly usable.

8.1.3 What is the effect of modelling an evolving CIG using *FCIG* in comparison with the HL7 standard for modelling CIGs?

FCIG was compared with the HL7 standard for modelling clinical guidelines Arden Syntax. The findings that answered this research question are synthesised through the discussion of its sub-questions that follow.

What is the effect of using FCIG, in comparison with Arden Syntax, on task time?

CIG modellers completed CIG encoding and maintenance tasks in less time when using *FCIG* in comparison with the existing HL7 standard.

What is the effect of using FCIG, in comparison with Arden Syntax, on success rate?

There was no difference in success rate when the CIG modellers completed tasks using *FCIG* as compared to when they used the HL7 standard. However, the CIG modellers completed the CIG modelling tasks with fewer errors when they used *FCIG*.

What is the effect of using FCIG, in comparison with Arden Syntax, on error rate?

There was no difference in error rate when CIG modellers completed CIG modelling tasks using *FCIG* in comparison with when they used Arden Syntax. In spite of this, CIG modellers completed with fewer number of errors when using *FCIG*.

What is the effect of using FCIG, in comparison with Arden Syntax, on the number of errors made?

CIG modellers completed CIG modelling tasks with fewer errors when using *FCIG* as in comparison to the existing HL7 standard.

What is the effect of using FCIG on efficiency?

CIG modellers completed CIG modelling tasks more efficiently when using *FCIG* in comparison with the HL7 standard.

What is the effect of using FCIG on LOC?

CIG modellers completed CIG modelling tasks with fewer lines of code when using *FCIG* in comparison with the HL7 standard.

What is the effect of using FCIG on ELOC?

CIG modellers completed CIG modelling tasks with fewer effective lines of code when using *FCIG* in comparison with the HL7 standard.

What is the effect of using FCIG on SUS scores?

CIG modellers perceived *FCIG* as a more user-friendly CIG modelling language as compared to the HL7 standard Arden Syntax.

8.2 Summary of contributions

This research explored the potential of managing CPG changes using a model-driven engineering approach in a clinical setting. In conducting these studies, I have established a base of knowledge for the development of evolving guideline-based clinical decision support tools that can be used to improve the quality of clinical care in low- and middle-income countries.

8.2.1 Evolving clinical practice guideline formalization

CPGs are formalized into CIGs to increase adoption and application of CPGs in computer-supported clinical settings [49, 311]. But CPGs evolve over time as the science and technology behind clinical practice improves, necessitating changes in their formalized form as CIGs. This gives rise to the question: How should evolving CPGs be formalized to facilitate their integration into Electronic Medical Record systems? Hence evolving CPG formalization forming the underlying theoretical perspective for this work.

There are a number of studies that have proposed various approaches to CPG formalization [11]. Attempts have been made to address the complexities involved when supporting evolving CPGs in CPG formalization systems. For example, Miller et al. [115] focused on maintaining and revalidating child immunization CIGs when changes occur. Whilst the other studies were concerned with supporting multiple versions of a CIG in a single repository [11, 61, 118]. As such, little emphasis was placed on how to support the CIG modeller to effect a change in a CIG when the change occurs, even more so, in low- and middle-income countries. This work seeks to fill this gap.

8.2.2 Characterisation of CPG changes

In this research, a characterisation of CPG changes was proposed by looking at semantic elements that are affected when changes occur. The characteristics of the CPG changes were elicited through an inductive learning approach that used successive sets of CPG documents obtained from the Malawi Ministry of Health. In addition, this work looked at CPG representation primitives in current executable CIG modelling languages and identified semantic elements that are typically affected by CPG changes but are not currently supported by explicit CPG modelling primitives in current CIG modelling languages.

8.2.3 Four-layer computer-interpretable guideline modelling architecture

In this work, a novel layered-architecture for modelling evolving CIGs was designed using a model-driven approach. The novel CIG modelling architecture is based on the four-layer Model-Driven Architecture (MDA). Further to that, this novel CIG modelling architecture was implemented and evaluated using open standards. Each layer of the architecture focused on a different aspect of CIG modelling concerns.

8.2.4 A compact and usable CIG modelling language

This work introduced a fine-grained yet a compact CIG modelling language that was named *FCIG*. Modelling CIGs using CIG modelling methods published in current literature can entail quite a significant effort [62]. Further to that, understanding the semantics of some existing CIG modelling languages prove difficult to a CIG modeller, increasing the likelihood of introducing errors during the CIG modelling process [62, 102, 312]. The novel CIG modelling language has formal semantics that are fully compatible with the layered CIG modelling architecture. The feasibility of modelling CIGs using *FCIG* was evaluated using Ecore and the Eclipse Modelling Framework (EMF) where an Eclipse IDE plugin was implemented with full language support and smart-editing features for *FCIG*.

8.2.5 Contributions to Health Informatics and Development

This work contributes to Health Informatics research in low- and middle-income countries by enhancing the understanding of requirements for CPG formalization in such settings. Furthermore, this work identifies a systematic approach to developing and evaluating interoperable evolving-CIG formalization tools for clinical decision support in low-resource settings.

8.3 Limitations of the study

There are a number of important limitations that need to be highlighted in this work. Whilst the results in this work are grounded in significant research effort and some of the evaluations were done using CIG modelling experts, the foundational aspects of this work have not been evaluated in a production environment and over a significant period of time. A necessary and critical step would be to evaluate the novel CIG modelling architecture with CIG modellers in a production environment and over a significant length of time to evaluate its impact in a real clinical practice setting. The second limitation is that the characteristics of CPG changes may not be exhaustive. Hence, we may not know further characteristics of CPG changes until the CIG modelling architecture has been used successfully in practice and over a significant period of time. The third limitation is that this work was confined to CPG formalization for electronic medical record systems in Malawi. Though the CPGs that were used in this work have been adapted from the World Health Organization (WHO) CPGs, which have also been adopted in other low- and middle-income countries, the findings from this work cannot be generalised to high-income countries. Going forward, the CIG modelling approach should be evaluated in different settings to overcome this limitation.

8.4 Opportunities for future work

There are a number of potential areas that would be interesting to pursue in the future. The subsections that follow discuss these areas.

8.4.1 Deployment in a real-world clinical setting

The evolving CIG modelling architecture and its related CIG modelling language *FCIG* has been evaluated through prototype experimentation. It would be ideal to evaluate *FCIG* and the CIG modelling architecture from this work in a real-world clinical setting.

8.4.2 Co-designing with clinical personnel

This work can be extended to design and build more CIG modelling tools with clinical personnel in addition to the CIG modellers who are clinical knowledge engineering experts. Such tools would be ideal as they would enable clinical experts to directly author and verify CIGs using the proposed CIG modelling tools.

8.4.3 Using formal CIGs in guideline adherence interventions

The CIG modelling architecture presented in this work can be extended with CPG adherence measurement algorithms. Such functionality can be used for measuring and subsequent reporting of clinical performance to health managers.

8.4.4 Extension of *FCIG*

Though the results from this work show that *FCIG* is more effective than the HL7 standard Arden Syntax when modelling evolving CIGs, the language should be evaluated in practice and possibly through a longitudinal study. And from the results of such an evaluation, more advanced language constructs such as logical connectors or negation operators can be added to *FCIG* to cater for advanced CIG modellers.

8.4.5 Evaluation with other upcoming CIG modelling standards

There are other upcoming standards for modelling CIGs such as the Guideline Definition Language of the openEHR [313]. *FCIG* should be evaluated against any other upcoming standards for modelling CIGs to ensure its relevancy in addition to incorporating any relevant new features as the language evolves.

8.4.6 Evaluating the approach in a longitudinal study

The evaluation in this work was mostly carried out in an experimental setting. It would be ideal to evaluate the CIG modelling architecture in a longitudinal study to measure its impact in practice.

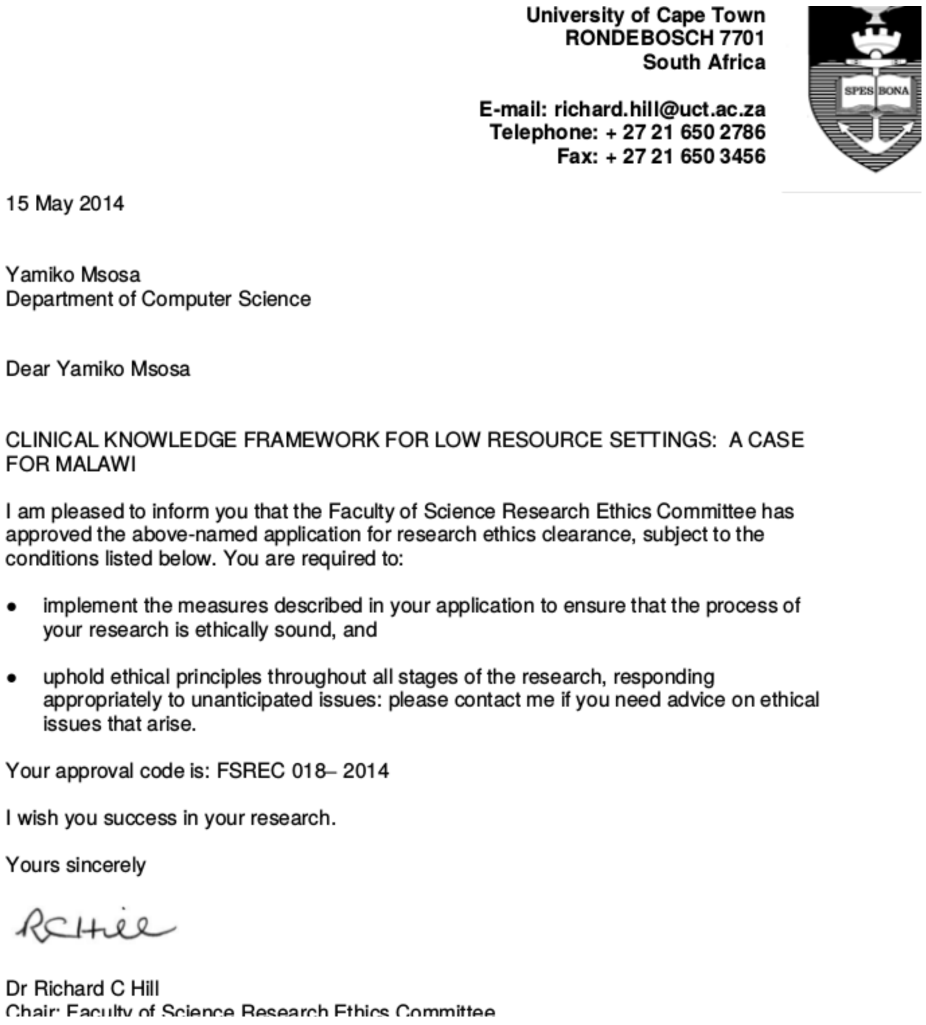
8.5 Final remarks

This work has the potential to significantly impact the deployment of sustainable clinical decision support systems that require up-to-date clinical practice guidelines. If the novel approach and its related framework is adopted in practice, it could improve the tool base that supports task-shifted health workers, thereby enabling patients to receive better care in low- and middle-income countries which shoulder the majority of the global disease burden.

Appendix A

Legal and ethical clearances

A.1 UCT ethical clearance approval



University of Cape Town
RONDEBOSCH 7701
South Africa

E-mail: richard.hill@uct.ac.za
Telephone: + 27 21 650 2786
Fax: + 27 21 650 3456

15 May 2014

Yamiko Msosa
Department of Computer Science

Dear Yamiko Msosa

CLINICAL KNOWLEDGE FRAMEWORK FOR LOW RESOURCE SETTINGS: A CASE FOR MALAWI

I am pleased to inform you that the Faculty of Science Research Ethics Committee has approved the above-named application for research ethics clearance, subject to the conditions listed below. You are required to:

- implement the measures described in your application to ensure that the process of your research is ethically sound, and
- uphold ethical principles throughout all stages of the research, responding appropriately to unanticipated issues: please contact me if you need advice on ethical issues that arise.

Your approval code is: FSREC 018– 2014

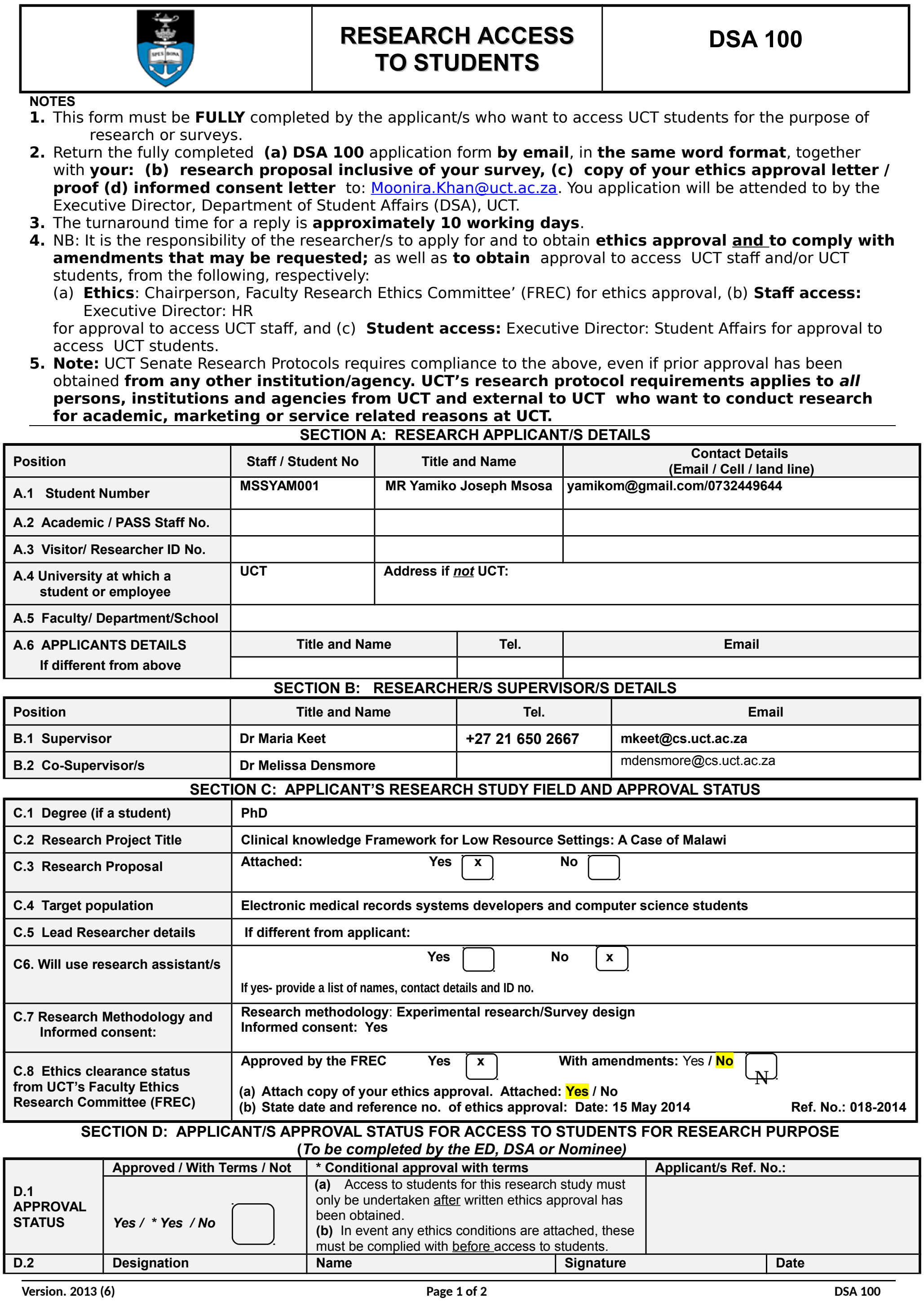
I wish you success in your research.

Yours sincerely

RCHill

Dr Richard C Hill
Chair: Faculty of Science Research Ethics Committee

A.2 UCT student access authorisation



	RESEARCH ACCESS TO STUDENTS	**DSA 100**

NOTES

1. This form must be **FULLY** completed by the applicant/s who want to access UCT students for the purpose of research or surveys.
2. Return the fully completed **(a) DSA 100** application form **by email**, in **the same word format**, together with **your: (b) research proposal inclusive of your survey, (c) copy of your ethics approval letter / proof (d) informed consent letter** to: Moonira.Khan@uct.ac.za. You application will be attended to by the Executive Director, Department of Student Affairs (DSA), UCT.
3. The turnaround time for a reply is **approximately 10 working days**.
4. NB: It is the responsibility of the researcher/s to apply for and to obtain **ethics approval <u>and</u> to comply with amendments that may be requested;** as well as **to obtain** approval to access UCT staff and/or UCT students, from the following, respectively:
   (a) **Ethics**: Chairperson, Faculty Research Ethics Committee' (FREC) for ethics approval, (b) **Staff access:** Executive Director: HR
   for approval to access UCT staff, and (c) **Student access:** Executive Director: Student Affairs for approval to access UCT students.
5. **Note:** UCT Senate Research Protocols requires compliance to the above, even if prior approval has been obtained **from any other institution/agency. UCT's research protocol requirements applies to *all* persons, institutions and agencies from UCT and external to UCT who want to conduct research for academic, marketing or service related reasons at UCT.**

SECTION A: RESEARCH APPLICANT/S DETAILS

Position	Staff / Student No	Title and Name	Contact Details (Email / Cell / land line)
A.1 Student Number	MSSYAM001	MR Yamiko Joseph Msosa	yamikom@gmail.com/0732449644
A.2 Academic / PASS Staff No.			
A.3 Visitor/ Researcher ID No.			
A.4 University at which a student or employee	UCT	Address if *not* UCT:	
A.5 Faculty/ Department/School			
A.6 APPLICANTS DETAILS If different from above	Title and Name	Tel.	Email

SECTION B: RESEARCHER/S SUPERVISOR/S DETAILS

Position	Title and Name	Tel.	Email
B.1 Supervisor	Dr Maria Keet	+27 21 650 2667	mkeet@cs.uct.ac.za
B.2 Co-Supervisor/s	Dr Melissa Densmore		mdensmore@cs.uct.ac.za

SECTION C: APPLICANT'S RESEARCH STUDY FIELD AND APPROVAL STATUS

C.1 Degree (if a student)	PhD
C.2 Research Project Title	Clinical knowledge Framework for Low Resource Settings: A Case of Malawi
C.3 Research Proposal	Attached: Yes ☒ No □
C.4 Target population	Electronic medical records systems developers and computer science students
C.5 Lead Researcher details	If different from applicant:
C6. Will use research assistant/s	Yes □ No ☒ If yes- provide a list of names, contact details and ID no.
C.7 Research Methodology and Informed consent:	Research methodology: Experimental research/Survey design Informed consent: Yes
C.8 Ethics clearance status from UCT's Faculty Ethics Research Committee (FREC)	Approved by the FREC Yes ☒ With amendments: Yes / [✓ No] N (a) Attach copy of your ethics approval. Attached: [✓ Yes] / No (b) State date and reference no. of ethics approval: Date: 15 May 2014 Ref. No.: 018-2014

SECTION D: APPLICANT/S APPROVAL STATUS FOR ACCESS TO STUDENTS FOR RESEARCH PURPOSE
(*To be completed by the ED, DSA or Nominee)*

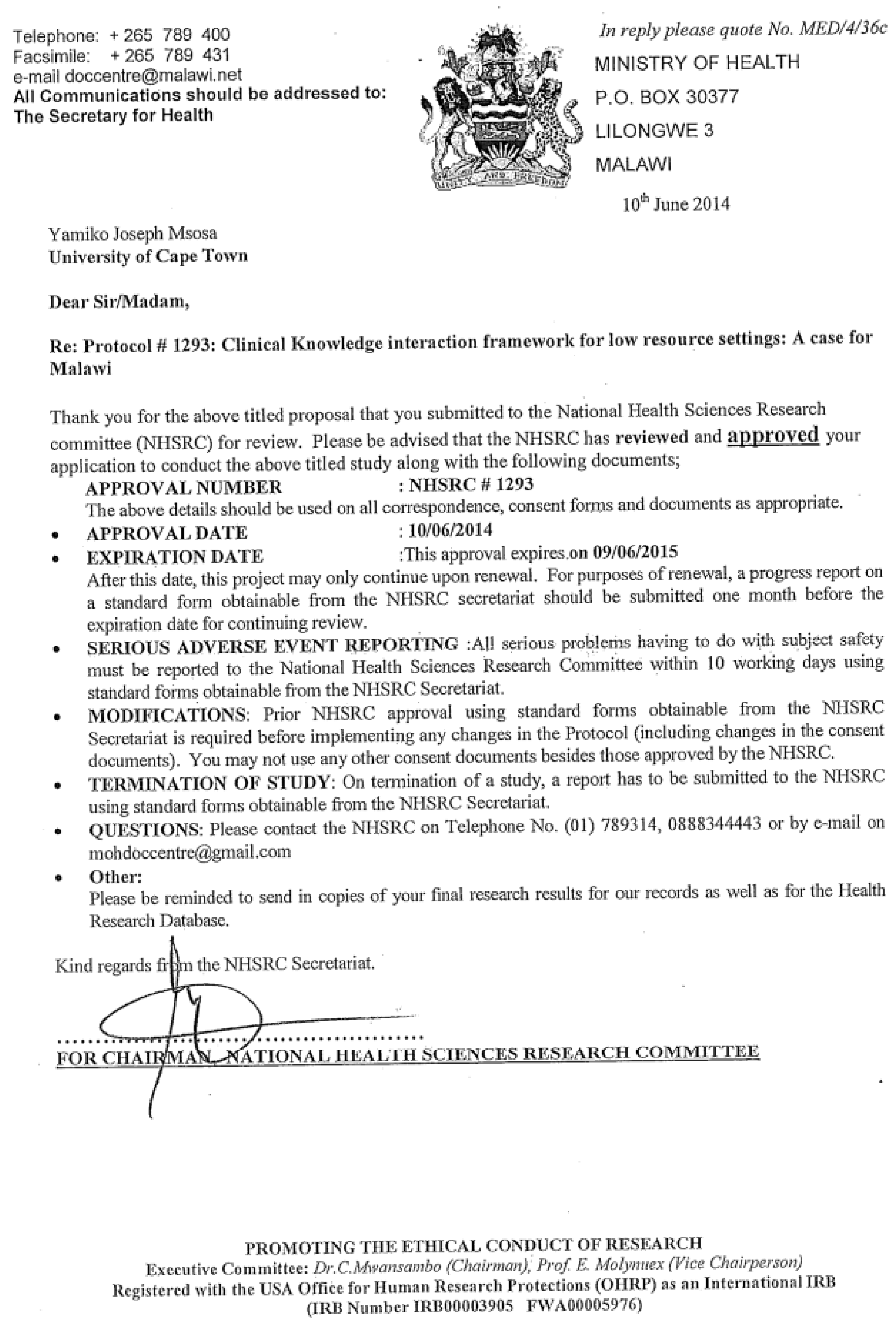
	Approved / With Terms / Not	* Conditional approval with terms	Applicant/s Ref. No.:
D.1 APPROVAL STATUS	*Yes / * Yes / No* □	**(a)** Access to students for this research study must only be undertaken <u>after</u> written ethics approval has been obtained. **(b)** In event any ethics conditions are attached, these must be complied with <u>before</u> access to students.	

D.2	Designation	Name	Signature	Date

Version. 2013 (6) Page 1 of 2 DSA 100

APPROVED BY:	*Executive Director Department of Student Affairs*			

A.3 Malawi Ministry of Health ethical clearance



Telephone: + 265 789 400
Facsimile: + 265 789 431
e-mail doccentre@malawi.net
All Communications should be addressed to:
The Secretary for Health

In reply please quote No. MED/4/36c
MINISTRY OF HEALTH
P.O. BOX 30377
LILONGWE 3
MALAWI

10th June 2014

Yamiko Joseph Msosa
University of Cape Town

Dear Sir/Madam,

Re: Protocol # 1293: Clinical Knowledge interaction framework for low resource settings: A case for Malawi

Thank you for the above titled proposal that you submitted to the National Health Sciences Research committee (NHSRC) for review. Please be advised that the NHSRC has **reviewed** and **approved** your application to conduct the above titled study along with the following documents;

APPROVAL NUMBER : **NHSRC # 1293**
The above details should be used on all correspondence, consent forms and documents as appropriate.

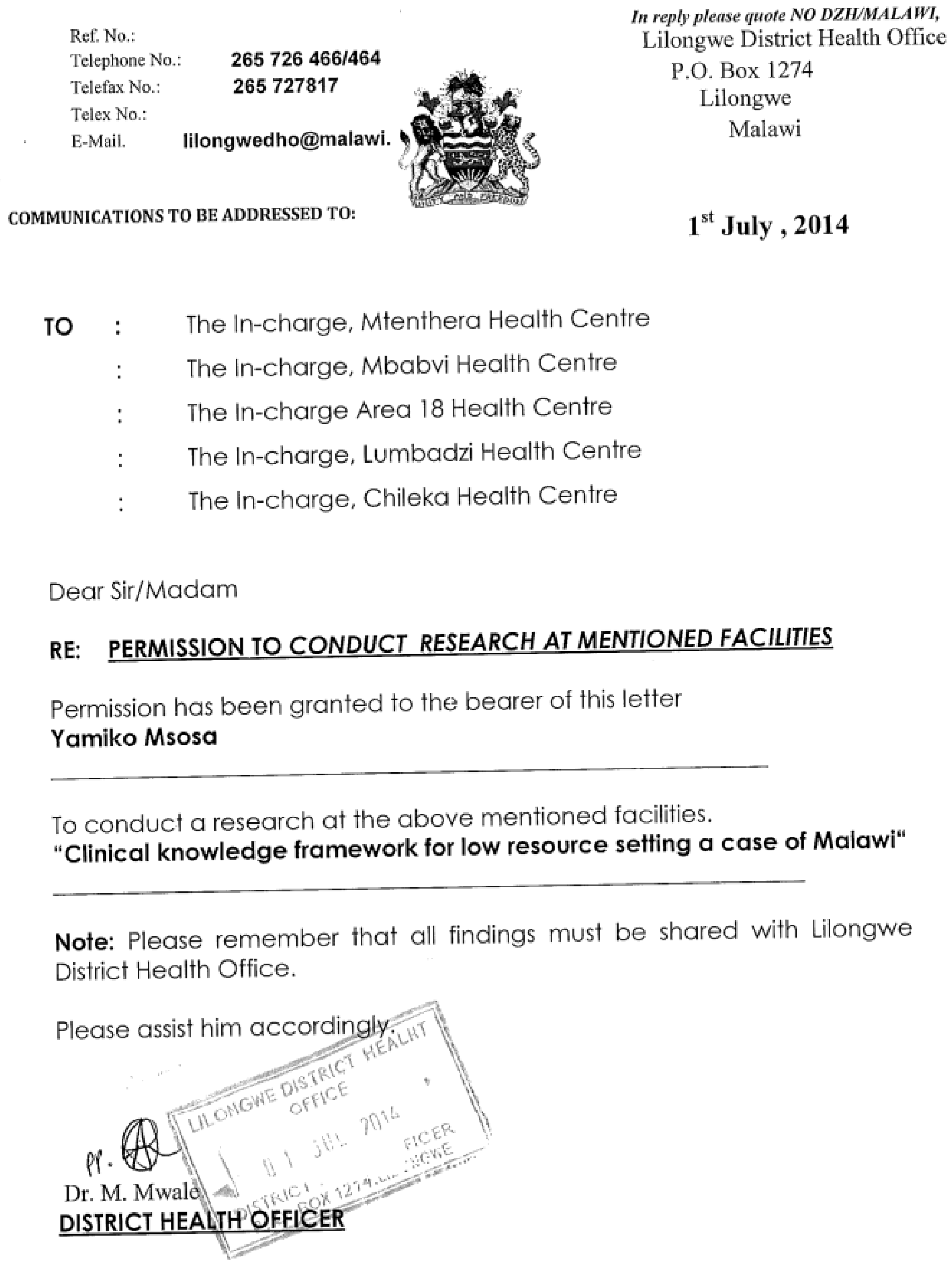
- **APPROVAL DATE** : **10/06/2014**
- **EXPIRATION DATE** :This approval expires **on 09/06/2015**
After this date, this project may only continue upon renewal. For purposes of renewal, a progress report on a standard form obtainable from the NHSRC secretariat should be submitted one month before the expiration date for continuing review.
- **SERIOUS ADVERSE EVENT REPORTING** :All serious problems having to do with subject safety must be reported to the National Health Sciences Research Committee within 10 working days using standard forms obtainable from the NHSRC Secretariat.
- **MODIFICATIONS**: Prior NHSRC approval using standard forms obtainable from the NHSRC Secretariat is required before implementing any changes in the Protocol (including changes in the consent documents). You may not use any other consent documents besides those approved by the NHSRC.
- **TERMINATION OF STUDY**: On termination of a study, a report has to be submitted to the NHSRC using standard forms obtainable from the NHSRC Secretariat.
- **QUESTIONS**: Please contact the NHSRC on Telephone No. (01) 789314, 0888344443 or by e-mail on mohdoccentre@gmail.com
- **Other:**
Please be reminded to send in copies of your final research results for our records as well as for the Health Research Database.

Kind regards from the NHSRC Secretariat.

....
FOR CHAIRMAN, NATIONAL HEALTH SCIENCES RESEARCH COMMITTEE

PROMOTING THE ETHICAL CONDUCT OF RESEARCH
Executive Committee: *Dr.C.Mwansambo (Chairman), Prof. E. Molynuex (Vice Chairperson)*
Registered with the USA Office for Human Research Protections (OHRP) as an International IRB
(IRB Number IRB00003905 FWA00005976)

A.4 Malawi Ministry of Health health centre access approval



Ref. No.:
Telephone No.: **265 726 466/464**
Telefax No.: **265 727817**
Telex No.:
E-Mail. **lilongwedho@malawi.**

In reply please quote NO DZH/MALAWI,
Lilongwe District Health Office
P.O. Box 1274
Lilongwe
Malawi

COMMUNICATIONS TO BE ADDRESSED TO:

1st July , 2014

TO : The In-charge, Mtenthera Health Centre
: The In-charge, Mbabvi Health Centre
: The In-charge Area 18 Health Centre
: The In-charge, Lumbadzi Health Centre
: The In-charge, Chileka Health Centre

Dear Sir/Madam

RE: PERMISSION TO CONDUCT RESEARCH AT MENTIONED FACILITIES

Permission has been granted to the bearer of this letter
Yamiko Msosa

To conduct a research at the above mentioned facilities.
"Clinical knowledge framework for low resource setting a case of Malawi"

Note: Please remember that all findings must be shared with Lilongwe District Health Office.

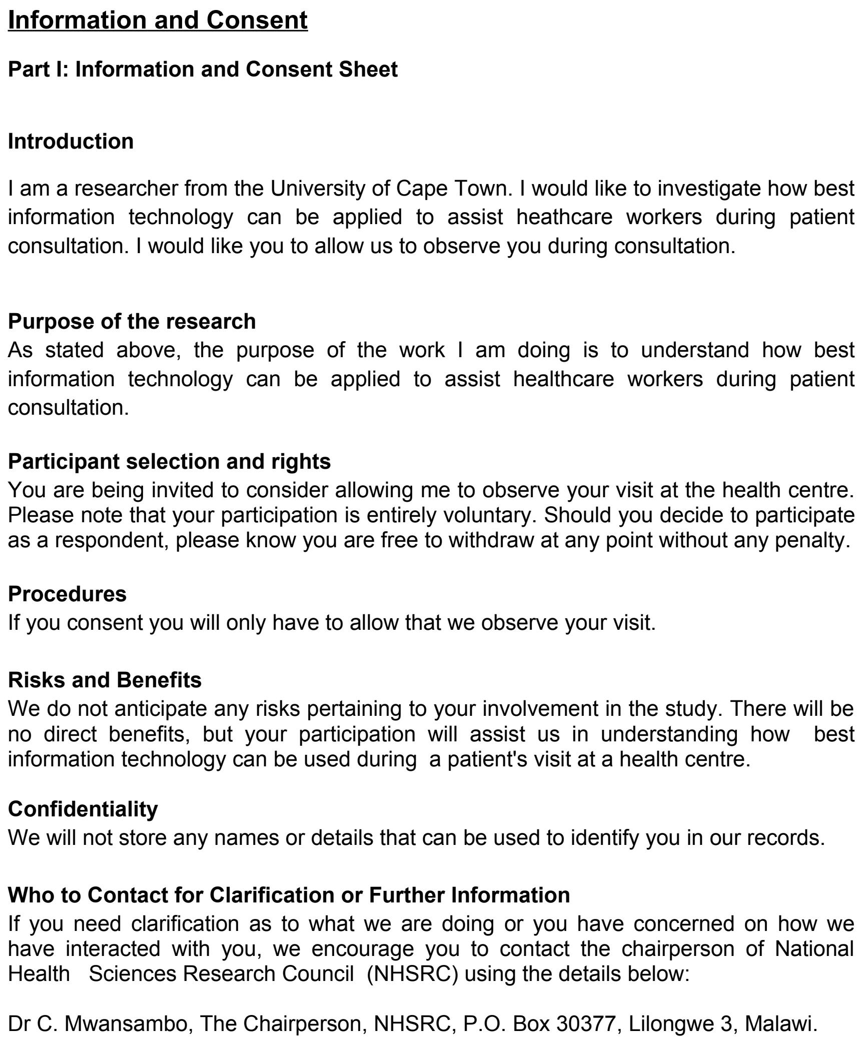
Please assist him accordingly.

LILONGWE DISTRICT HEALTH OFFICE
01 JUL 2014
DISTRICT [illegible] OFFICER
BOX 1274, [illegible] LILONGWE

pp.

Dr. M. Mwale
DISTRICT HEALTH OFFICER

A.5 Consent forms - patients



Information and Consent

Part I: Information and Consent Sheet

Introduction

I am a researcher from the University of Cape Town. I would like to investigate how best information technology can be applied to assist heathcare workers during patient consultation. I would like you to allow us to observe you during consultation.

Purpose of the research

As stated above, the purpose of the work I am doing is to understand how best information technology can be applied to assist healthcare workers during patient consultation.

Participant selection and rights

You are being invited to consider allowing me to observe your visit at the health centre. Please note that your participation is entirely voluntary. Should you decide to participate as a respondent, please know you are free to withdraw at any point without any penalty.

Procedures

If you consent you will only have to allow that we observe your visit.

Risks and Benefits

We do not anticipate any risks pertaining to your involvement in the study. There will be no direct benefits, but your participation will assist us in understanding how best information technology can be used during a patient's visit at a health centre.

Confidentiality

We will not store any names or details that can be used to identify you in our records.

Who to Contact for Clarification or Further Information

If you need clarification as to what we are doing or you have concerned on how we have interacted with you, we encourage you to contact the chairperson of National Health Sciences Research Council (NHSRC) using the details below:

Dr C. Mwansambo, The Chairperson, NHSRC, P.O. Box 30377, Lilongwe 3, Malawi.

Part II: Certificate of Consent

I have been invited to participate in the technology assisted patient consultation study. I have read the foregoing information (The foregoing has been read to me). I have had the opportunity to ask questions about it and any questions I have asked have been answered. My participation in the study is entirely voluntary.

☐ I consent to participate in the study
☐ I do not consent

_____________________ __________________________ ______________
Participant Name Participant Signature Date

_____________________ __________________________ ______________
Observer Observer Signature Date

Chidziwitso ndi Chilolezo (Odwala)

Ndondomeko yoyamba: Chidziwitso

Mawu otsogolera

Ine ndi wophunzira ku sukulu ya ukachenjede ya University of Cape Town. Ndikufuna kuti ndifufuze njira zabwino zogwiritsa ntchito luso la makono la mauthenga ndi kulumikizana pothandizira a chipatala pamene akuthandiza odwala. Ndikufuka kuti mundilore kuti ndiwonelere pamene mukuthandizidwa ndi a chipatala.

Zifukwa za kafukufuku
Monga tanenera, ntchito imene tikupangayi ithandiza kuti tidziwe motsindika njira zabwino zogwiritsira luso la makono la mauthenga ndi kulumikizana pothandizira a chipatala pamene akuthandiza odwala.

Kasankhidwe ndi ufulu wa ofunsidwa
Tikupempha kuti mutilore kuti ndikufunseni mafunso ndi kuwonelera mukuthandizidwa ndi a chipatala. Dziwani kuti kuthandiza kwanu ndi kosakakamizidwa. Ngati mutalora kuti mulowe nawo mu kafukufuku ameneyi, mudziwe kuti muli ndi ufulu otuluka nthawi iliyonse popanda chilango.

Ndondomeko
Ngati mutalora kuti mulowe nawo mu kafukufuku ameneyi, mukulora kuti tiwonelere pokhapo pamene mukuthandizidwa ndi a chipatala.

Chiopsezo ndi ubwino
Palibe choopsa chilichonse chimene chingadze chifukwa cha kulowa kwanu mu kafukufuku ameneyi. Palibe malipiro mukalowa kafukufuku ameneyi, koma kulowa kwanu kutithandiza kuti tidziwe motsindika za njira zabwino zogwiritsa ntchito luso la makono la mauthenga ndi kulumikizana pothandizira a chipatala pamene akuthandiza odwala pa chipatala.

Chinsisi
Sitisunga mayina kapena chilichonse mu kawundula wathu chimene chingathe kulondolera wina aliyense kwa kwa inu.

Amene mungathe kulumizana nawo pofuna kudziwa zambiri
Ngati mukufuna kudziwa zambiri za kafukufuku ameneyi kapena muli ndi nkhawa ndi m'mene tachezera, mukhoza kulumikizana ndi wa pampando wa bungwe loyang'anira kafukufuku wa za umoyo la National Health Sciences Research Council (NHSRC) motere: **Dr C. Mwansambo, The Chairperson, NHSRC, P.O. Box 30377, Lilongwe 3, Malawi.**

Ndondomeko yachiwiri: Chilolezo

Ndayitanidwa kuti ndilowe nawo kafukufuku wa luso la makono la mauthenga ndi kulumikizana mu chipatala. Ndawelenga uthenga umene walembedwa kale (andiwelengera uthenga umene walembedwa kale). Ndinali ndi mpata wofunsa mafunso ndipo mafunso anga onse ayankhidwa. Kutenga mbali kwanga mu kafukufukuyi ndi kosakakamizidwa.

☐ Ndikulora kuti ndilowe nawo mu kafukufukuyi
☐ Sindikulora kuti ndilowe nawo mu kafukufukuyi

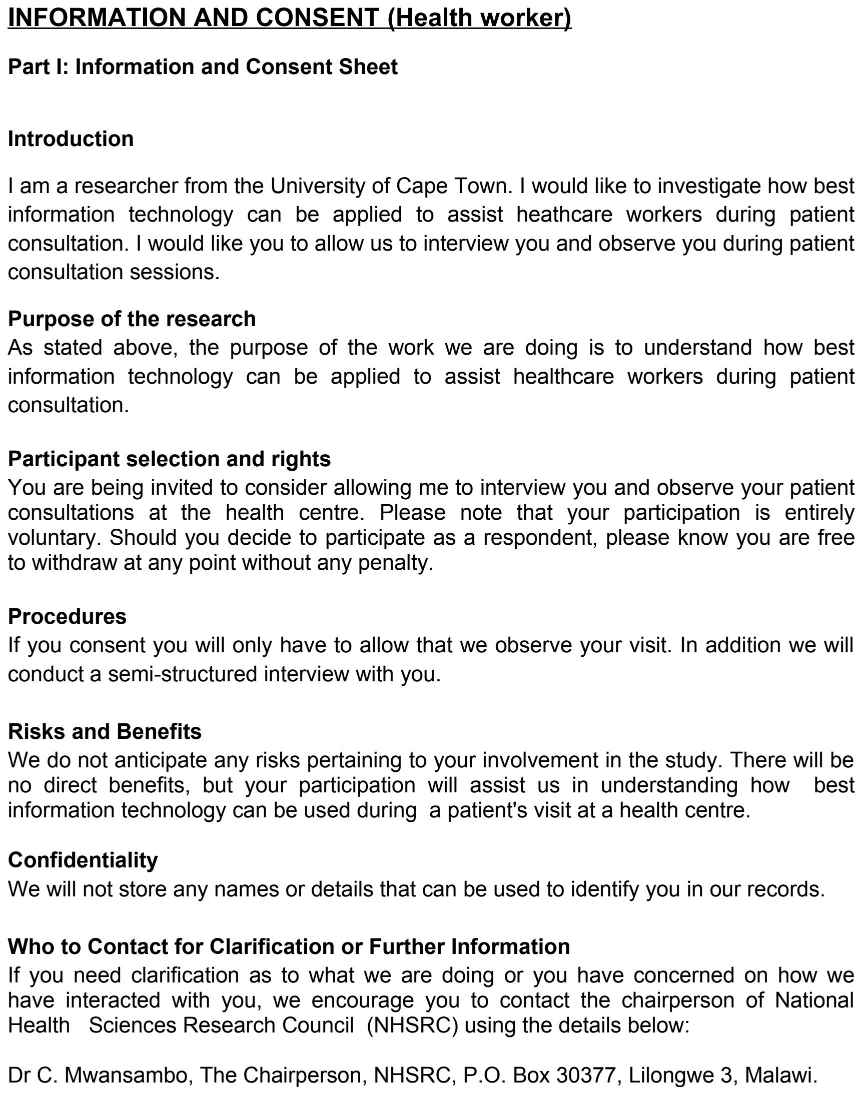
______________________ ___________________________ ______________

Dzina la ofunsidwa Chidindo cha ofunsidwa Tsiku

______________________ ___________________________ ______________

Ofunsa Chidindo cha ofunsa Tsiku

A.6 Consent forms - healthworkers



INFORMATION AND CONSENT (Health worker)

Part I: Information and Consent Sheet

Introduction

I am a researcher from the University of Cape Town. I would like to investigate how best information technology can be applied to assist heathcare workers during patient consultation. I would like you to allow us to interview you and observe you during patient consultation sessions.

Purpose of the research

As stated above, the purpose of the work we are doing is to understand how best information technology can be applied to assist healthcare workers during patient consultation.

Participant selection and rights

You are being invited to consider allowing me to interview you and observe your patient consultations at the health centre. Please note that your participation is entirely voluntary. Should you decide to participate as a respondent, please know you are free to withdraw at any point without any penalty.

Procedures

If you consent you will only have to allow that we observe your visit. In addition we will conduct a semi-structured interview with you.

Risks and Benefits

We do not anticipate any risks pertaining to your involvement in the study. There will be no direct benefits, but your participation will assist us in understanding how best information technology can be used during a patient's visit at a health centre.

Confidentiality

We will not store any names or details that can be used to identify you in our records.

Who to Contact for Clarification or Further Information

If you need clarification as to what we are doing or you have concerned on how we have interacted with you, we encourage you to contact the chairperson of National Health Sciences Research Council (NHSRC) using the details below:

Dr C. Mwansambo, The Chairperson, NHSRC, P.O. Box 30377, Lilongwe 3, Malawi.

Part II: Certificate of Consent

I have been invited to participate in the technology assisted patient consultation study. I have read the foregoing information (The foregoing has been read to me). I have had the opportunity to ask questions about it and any questions I have asked have been answered. My participation in the study is entirely voluntary.
☐ I consent to participate in the study
☐ I do not consent

_____________________ __________________________ ______________

Participant Name Participant Signature Date

_____________________ __________________________ ______________

Interviewer Interviwer Signature Date

Chidziwitso ndi Chilolezo (Ogwira ntchito pa za umoyo)

Ndondomeko yoyamba: Chidziwitso

Mawu otsogolera

Ine ndi wophunzira ku sukulu ya ukachenjede ya University of Cape Town. Ndikufuna kuti ndifufuze njira zabwino zogwiritsa ntchito luso la makono la mauthenga ndi kulumikizana pothandizira a chipatala pamene akuthandiza odwala. Ndikufuka kuti mundilore kuti ndikufunseni mafunso ndi kuwonelera pamene mukuthandiza odwala.

Zifukwa za kafukufuku

Monga tanenera, ntchito imene tikupangayi ithandiza kuti tidziwe motsindika za njira zabwino zogwiritsira luso la makono la mauthenga ndi kulumikizana pothandizira a chipatala pamene akuthandiza odwala.

Kasankhidwe ndi ufulu wa ofunsidwa

Tikupempha kuti mutilore kuti ndikufunseni mafunso ndi kuwonelera mukuthandiza odwala pa chipatalachi. Dziwani kuti kuthandiza kwanu ndi kosakakamizidwa. Ngati mutalora kuti mulowe nawo mu kafukufuku ameneyi, mudziwe kuti muli ndi ufulu otuluka nthawi ina iliyonse popanda chilango.

Ndondomeko

Ngati mutalora kuti mulowe nawo mu kafukufuku ameneyi, mukulora kuti tiwonelere pokhapo pamene mukuthandiza odwala. Moonjezera, tidzakufunsaniko mafunso pa nthawi ina imene tingagwirizane.

Chiopsezo ndi ubwino

Palibe choopsa chilichonse chimene chingadze chifukwa cha kulowa kwanu mu kafukufuku ameneyi. Palibe malipiro mukalowa kafukufuku ameneyi, koma kulowa kwanu kutithandiza ife kuti tidziwe motsindika za njira zabwino zogwiritsa ntchito luso la makono la mauthenga ndi kulumikizana pothandizira a chipatala pamene akuthandiza odwala pa chipatala.

Chinsisi

Sitisunga mayina kapena chilichonse mu kawundula wathu chimene chingathe kulondolera wina aliyense kwa kwa inu.

Amene mungathe kulumizana nawo pofuna kudziwa zambiri

Ngati mukufuna kudziwa zambiri za kafukufuku ameneyi kapena muli ndi nkhawa ndi m'mene tachezera, mukhoza kulumikizana ndi wa pampando wa bungwe loyang'anira kafukufuku wa za umoyo la National Health Sciences Research Council (NHSRC) motere: **Dr C. Mwansambo, The Chairperson, NHSRC, P.O. Box 30377, Lilongwe 3, Malawi.**

Ndondomeko yachiwiri: Chilolezo

Ndapemphedwa kuti ndilowe nawo kafukufuku wa luso la makono la mauthenga ndi kulumikizana pa chipatala. Ndawelenga uthenga umene walembedwa kale (andiwelengera uthenga umene walembedwa kale). Ndinali ndi mpata wofunsa mafunso ndipo mafunso anga onse ayankhidwa. Kutenga mbali kwanga mu kafukufukuyi ndi kosakakamizidwa.

☐ Ndikulora kuti ndilowe nawo mu kafukufuku ameneyi
☐ Sindikulora kuti ndilowe nawo mu kafukufuku ameneyi

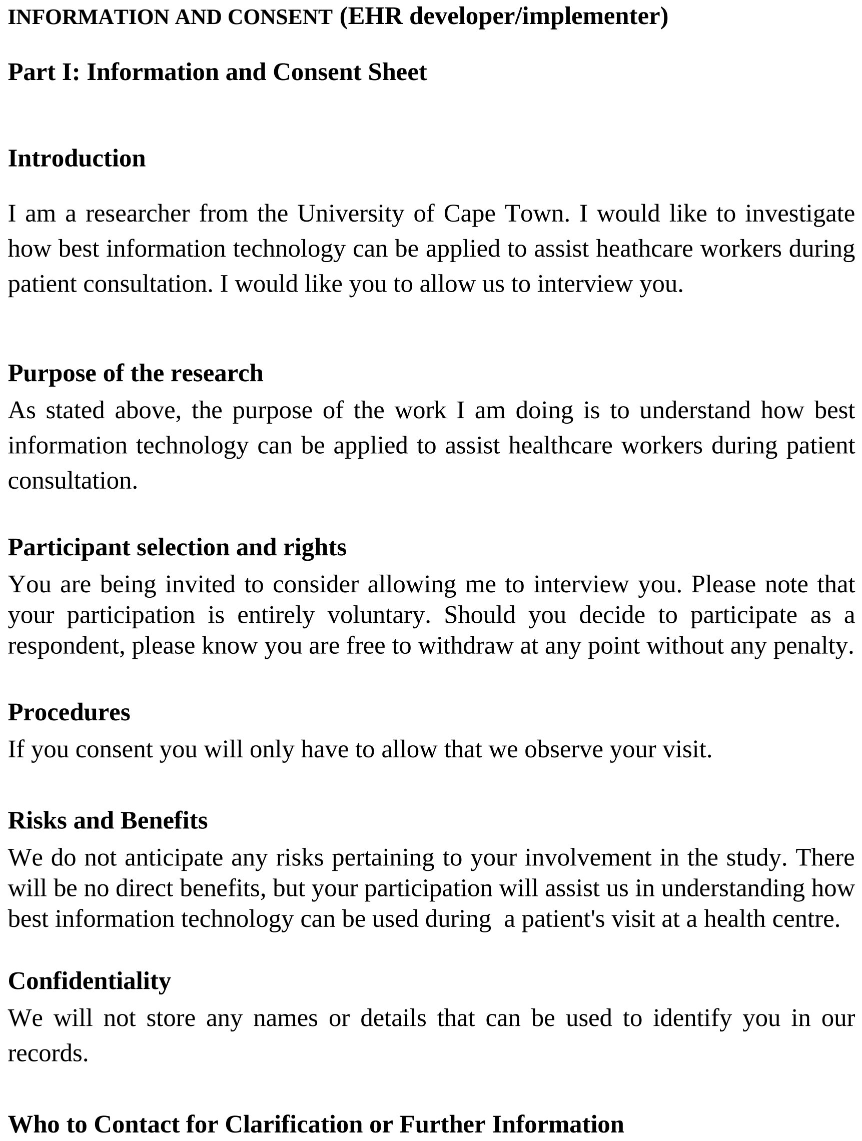
____________________   ________________________   ______________

Dzina la ofunsidwa   Chidindo cha ofunsidwa   Tsiku

____________________   ________________________   ______________

Ofunsa   Chidindo cha ofunsa   Tsiku

A.7 Consent forms - software developers



INFORMATION AND CONSENT (EHR developer/implementer)

Part I: Information and Consent Sheet

Introduction

I am a researcher from the University of Cape Town. I would like to investigate how best information technology can be applied to assist heathcare workers during patient consultation. I would like you to allow us to interview you.

Purpose of the research

As stated above, the purpose of the work I am doing is to understand how best information technology can be applied to assist healthcare workers during patient consultation.

Participant selection and rights

You are being invited to consider allowing me to interview you. Please note that your participation is entirely voluntary. Should you decide to participate as a respondent, please know you are free to withdraw at any point without any penalty.

Procedures

If you consent you will only have to allow that we observe your visit.

Risks and Benefits

We do not anticipate any risks pertaining to your involvement in the study. There will be no direct benefits, but your participation will assist us in understanding how best information technology can be used during a patient's visit at a health centre.

Confidentiality

We will not store any names or details that can be used to identify you in our records.

Who to Contact for Clarification or Further Information

If you need clarification as to what we are doing or you have concerned on how we have interacted with you, we encourage you to contact the chairperson of National Health Sciences Research Council (NHSRC) using the details below:

Dr C. Mwansambo, The Chairperson, NHSRC, P.O. Box 30377, Lilongwe 3, Malawi.

Part II: Certificate of Consent

I have been invited to participate in the technology assisted patient consultation study. I have read the foregoing information (The foregoing has been read to me). I have had the opportunity to ask questions about it and any questions I have asked have been answered. My participation in the study is entirely voluntary.

□ I consent to participate in the study
□ I do not consent

______________________ ___________________________ _______________
Participant Name Participant Signature Date

______________________ ___________________________ _______________
Interviewer Interviewer Signature Date

Appendix B

The concrete syntax using Xtext grammar language

```
grammar org.xtext.dsl.CIG
hidden(WS, ML_COMMENT, SL_COMMENT)
import "http://www.eclipse.org/emf/2002/Ecore" as ecore
generate cig "http://www.xtext.org/dsl/CIG"

Guideline:
    statements+=(Condition | Action)*
    statements+=(Recommendation)*
;

Recommendation:
    'Recommendation' name=ID ':'
        'Conditions' conditions+=[Condition](',' conditions+=[Condition])*
        'Actions' actions+=[Action](',' actions+=[Action])*
;

Condition:
    'Condition' name = ID ':'
         decisionVariable=DecisionVariable relator=Relator variableValue=
    ↪ VariableValue
         (unit=Unit)?
;

DecisionVariable:
    value = STRING |
    value = ID
;

Relator :
    value = 'is' |
    value = '=' |
    value = '>' |
    value = '>=' |
    value = '<' |
    value = '<='
;

VariableValue:
    value = NUMBER |
    value = STRING |
    value = 'true' |
    value = 'false' |
    value = ID
;

Unit:
    value = STRING | value = ID
;

Action:
    'Action' name = ID ':'
        actionVerb = ActionVerb actionVerbComplement = ActionVerbComplement
;

ActionVerb:
    value = ID |
    value = STRING

;

ActionVerbComplement:
    value = ID |
    value = STRING
;

/* Reusing common terminals from Xtext */
terminal ID    : '^'?('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'_'|'0'..'9')*;

terminal STRING    :
            '"' ( '\\' . /* 'b'|'t'|'n'|'f'|'r'|'u'|'"'|"'"|
            '\\' */ | !('\\'|'"') )* '"' |
            "'" ( '\\' . /* 'b'|'t'|'n'|'f'|'r'|'u'|'"'|"'"|
            '\\' */ | !('\\'|"'") )* "'"
        ;
terminal ML_COMMENT    : '/*' -> '*/';
terminal SL_COMMENT     : '//' !('\n'|'\r')* ('\r'? '\n')?;

terminal WS    : (' '|'\t'|'\r'|'\n')+;

/* Definition of a optionally signed number in the format -9999.999  */
terminal NUMBER returns ecore::EString:
    ('-')?('0'..'9')* ('.' ('0'..'9')+)?;
```

Appendix C

OWL specification for the CIG modelling language

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
   xml:base="http://www.yamiko.org/ontologies/cpg_model"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xml="http://www.w3.org/XML/1998/namespace"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   ontologyIRI="http://www.yamiko.org/ontologies/cpg_model">
  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <Declaration>
    <Class IRI="#Unit"/>
  </Declaration>
  <Declaration>
    <Class IRI="#VariableValue"/>
  </Declaration>
  <Declaration>
    <ObjectProperty IRI="#hasVariableValue"/>
  </Declaration>
  <Declaration>
    <ObjectProperty IRI="#hasAction"/>
  </Declaration>
  <Declaration>
    <Class IRI="#Recommendation"/>
  </Declaration>
  <Declaration>
    <ObjectProperty IRI="#referencesCondition"/>
  </Declaration>
  <Declaration>
    <Class IRI="#Relator"/>
```

```
</Declaration>
<Declaration>
  <ObjectProperty IRI="#referencesAction"/>
</Declaration>
<Declaration>
  <Class IRI="#DecisionVariable"/>
</Declaration>
<Declaration>
  <Class IRI="#ActionVerb"/>
</Declaration>
<Declaration>
  <Class IRI="#Guideline"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasRecommendation"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasUnit"/>
</Declaration>
<Declaration>
  <Class IRI="#Condition"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasCondition"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasDecisionVariable"/>
</Declaration>
<Declaration>
  <Class IRI="#ActionVerbComplement"/>
</Declaration>
<Declaration>
  <Class IRI="#Action"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasRelator"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasActionVerb"/>
</Declaration>
<Declaration>
  <Class IRI="#ConditionWithUnit"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasActionVerbComplement"/>
</Declaration>
<EquivalentClasses>
  <Class IRI="#Action"/>
  <ObjectIntersectionOf>
    <ObjectIntersectionOf>
      <ObjectAllValuesFrom>
        <ObjectProperty IRI="#hasActionVerb"/>
        <Class IRI="#ActionVerb"/>
      </ObjectAllValuesFrom>
      <ObjectExactCardinality cardinality="1">
```

```
          <ObjectProperty IRI="#hasActionVerb"/>
          <Class IRI="#ActionVerb"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasActionVerbComplement"/>
          <Class IRI="#ActionVerbComplement"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasActionVerbComplement"/>
          <Class IRI="#ActionVerbComplement"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
    </ObjectIntersectionOf>
  </EquivalentClasses>
  <EquivalentClasses>
    <Class IRI="#Condition"/>
    <ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasDecisionVariable"/>
          <Class IRI="#DecisionVariable"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasDecisionVariable"/>
          <Class IRI="#DecisionVariable"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasRelator"/>
          <Class IRI="#Relator"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasRelator"/>
          <Class IRI="#Relator"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasVariableValue"/>
          <Class IRI="#VariableValue"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasVariableValue"/>
          <Class IRI="#VariableValue"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
    </ObjectIntersectionOf>
  </EquivalentClasses>
  <EquivalentClasses>
    <Class IRI="#ConditionWithUnit"/>
    <ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasDecisionVariable"/>
          <Class IRI="#DecisionVariable"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasDecisionVariable"/>
          <Class IRI="#DecisionVariable"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasRelator"/>
          <Class IRI="#Relator"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasRelator"/>
          <Class IRI="#Relator"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasUnit"/>
          <Class IRI="#Unit"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasUnit"/>
          <Class IRI="#Unit"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasVariableValue"/>
          <Class IRI="#VariableValue"/>
        </ObjectAllValuesFrom>
        <ObjectExactCardinality cardinality="1">
          <ObjectProperty IRI="#hasVariableValue"/>
          <Class IRI="#VariableValue"/>
        </ObjectExactCardinality>
      </ObjectIntersectionOf>
    </ObjectIntersectionOf>
  </EquivalentClasses>
  <EquivalentClasses>
    <Class IRI="#Guideline"/>
    <ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectSomeValuesFrom>
          <ObjectProperty IRI="#hasAction"/>
          <Class IRI="#Action"/>
        </ObjectSomeValuesFrom>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasAction"/>
          <Class IRI="#Action"/>
        </ObjectAllValuesFrom>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
```

```
        <ObjectSomeValuesFrom>
          <ObjectProperty IRI="#hasCondition"/>
          <Class IRI="#Condition"/>
        </ObjectSomeValuesFrom>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasCondition"/>
          <Class IRI="#Condition"/>
        </ObjectAllValuesFrom>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectSomeValuesFrom>
          <ObjectProperty IRI="#hasRecommendation"/>
          <Class IRI="#Recommendation"/>
        </ObjectSomeValuesFrom>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#hasRecommendation"/>
          <Class IRI="#Recommendation"/>
        </ObjectAllValuesFrom>
      </ObjectIntersectionOf>
    </ObjectIntersectionOf>
  </EquivalentClasses>
  <EquivalentClasses>
    <Class IRI="#Recommendation"/>
    <ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectSomeValuesFrom>
          <ObjectProperty IRI="#referencesAction"/>
          <Class IRI="#Action"/>
        </ObjectSomeValuesFrom>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#referencesAction"/>
          <Class IRI="#Action"/>
        </ObjectAllValuesFrom>
      </ObjectIntersectionOf>
      <ObjectIntersectionOf>
        <ObjectSomeValuesFrom>
          <ObjectProperty IRI="#referencesCondition"/>
          <Class IRI="#Condition"/>
        </ObjectSomeValuesFrom>
        <ObjectAllValuesFrom>
          <ObjectProperty IRI="#referencesCondition"/>
          <Class IRI="#Condition"/>
        </ObjectAllValuesFrom>
      </ObjectIntersectionOf>
    </ObjectIntersectionOf>
  </EquivalentClasses>
  <DisjointClasses>
    <Class IRI="#Action"/>
    <Class IRI="#ActionVerb"/>
    <Class IRI="#ActionVerbComplement"/>
    <Class IRI="#Condition"/>
    <Class IRI="#DecisionVariable"/>
    <Class IRI="#Guideline"/>
    <Class IRI="#Recommendation"/>
    <Class IRI="#Relator"/>
    <Class IRI="#Unit"/>
    <Class IRI="#VariableValue"/>
  </DisjointClasses>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasAction"/>
    <Class IRI="#Guideline"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasActionVerb"/>
    <Class IRI="#Action"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasActionVerbComplement"/>
    <Class IRI="#Action"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasCondition"/>
    <Class IRI="#Guideline"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasDecisionVariable"/>
    <Class IRI="#Condition"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasRecommendation"/>
    <Class IRI="#Guideline"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasRelator"/>
    <Class IRI="#Condition"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasUnit"/>
    <Class IRI="#ConditionWithUnit"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasVariableValue"/>
    <Class IRI="#Condition"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#referencesAction"/>
    <Class IRI="#Recommendation"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#referencesCondition"/>
    <Class IRI="#Recommendation"/>
  </ObjectPropertyDomain>
  <ObjectPropertyRange>
    <ObjectProperty IRI="#hasAction"/>
    <Class IRI="#Action"/>
  </ObjectPropertyRange>
  <ObjectPropertyRange>
    <ObjectProperty IRI="#hasActionVerb"/>
    <Class IRI="#ActionVerb"/>
  </ObjectPropertyRange>
```

```
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasActionVerbComplement"/>
  <Class IRI="#ActionVerbComplement"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasCondition"/>
  <Class IRI="#Condition"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasDecisionVariable"/>
  <Class IRI="#DecisionVariable"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasRecommendation"/>
  <Class IRI="#Recommendation"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasRelator"/>
  <Class IRI="#Relator"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasUnit"/>
  <Class IRI="#Unit"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasVariableValue"/>
  <Class IRI="#VariableValue"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#referencesAction"/>
  <Class IRI="#Action"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#referencesCondition"/>
  <Class IRI="#Condition"/>
</ObjectPropertyRange>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#Action</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This
  ↪ class is adapted from the element Action of the Guideline Elements Model (
  ↪ GEM) and the equivalents of the action representation primitive in existing
  ↪  guideline modelling platforms such as Guideline Interchange Format (GLIF),
  ↪  Arden Syntax and Shareable and Active Guideline Environments (SAGE). The
  ↪ Action Element defines an appropriate activity to be carried out given the
  ↪ specific circumstances.</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#ActionVerb</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">Adapted
  ↪ from the Element ActionVerb of the Guideline Elements Model (GEM). The
  ↪ Action Element defines the word or phrase in a recommendation that
  ↪ expresses action.</Literal>
</AnnotationAssertion>
```

```
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#ActionVerbComplement</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">Adapted
  ↪ from the Element ActionVerb of the Guideline Elements Model (GEM). The
  ↪ Action Element defines the word or phrase in a recommendation that
  ↪ expresses action.</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#Condition</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This
  ↪ class is adapted from the equivalents of the condition representation
  ↪ primitive in existing guideline modelling platforms such as Guideline
  ↪ Interchange Format (GLIF), Arden Syntax and Shareable and Active Guideline
  ↪ Environments (SAGE). It defines the specific circumstances that a
  ↪ recommendation is applicable.</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#ConditionWithUnit</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This
  ↪ class defines a Condition that is qualified with a Unit.</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#DecisionVariable</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This
  ↪ class is adapted from the element DecisionVariable of the Guideline
  ↪ Elements Model (GEM) and the equivalents of the decision representation
  ↪ primitive in existing guideline modelling platforms such as Guideline
  ↪ Interchange Format (GLIF), Arden Syntax and Shareable and Active Guideline
  ↪ Environments (SAGE). The Decision Variable element defines a condition that
  ↪  must be tested to indicate the appropriateness of a conditional
  ↪ recommendation.</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#Recommendation</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This
  ↪ class is adapted from the element Recommendation of the Guideline Elements
  ↪ Model (GEM). The Recommendation Element defines the statement of
  ↪ appropriate practice and the conditions under which it is to be undertaken.
  ↪ </Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#Relator</IRI>
  <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">Adopted
  ↪ from the class gfo:Relator of the foundational ontology General Formal
  ↪ Ontology (GFO). It mediates how a VariableValue instance relates to a
  ↪ particular decisionVariable instance.</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
    <IRI>#Unit</IRI>
    <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This is
    ↪ modelled from the class Unit of the Systematized Nomenclature of Medicine -
    ↪  Clinical Terms (SNOMED-CT) that defines Unit as a Qualifier Value.</
    ↪ Literal>
  </AnnotationAssertion>
  <AnnotationAssertion>
    <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
    <IRI>#VariableValue</IRI>
    <Literal datatypeIRI="http://www.w3.org/2000/01/rdf-schema#Literal">This
    ↪ class is adapted from the element Value of the Guideline Elements Model (
    ↪ GEM) and the equivalents of the Decision representation primitive in
    ↪ existing guideline modelling platforms such as Guideline Interchange Format
    ↪  (GLIF), Arden Syntax and Shareable and Active Guideline Environments (SAGE
    ↪ ). The Value element defines a specified state of a decision variable.</
    ↪ Literal>
  </AnnotationAssertion>
</Ontology>
<!-- Generated by the OWL API (version 4.1.3.20151118-2017) https://github.com/
    ↪ owlcs/owlapi -->
```

Appendix D

Representative sample of guideline recommendations

Table D.1: Guideline recommendations stratified by guideline

Guideline	Recommendation
Determining priority for CD4 count testing	HIV-positive pregnant women
	patients in WHO Stage 2
	follow up every 6 months
Definition of ART eligibility	Adults - WHO Clinical Stage 2 with a total lymphocyte count $< 1200/\text{mm}^3$
	Children over the age of 18 months – Lymphocyte count below threshold (3 years to $<$ 5 yrs)
	Children over the age of 18 months - Lymphocyte count below threshold (¡ 3 yrs)
	PSHD - oral candidiasis and severe pneumonia
	Stopping ART and CPT at 18 months - If the test is negative - On ART
	Lymphocyte counts below threshold values for starting ART ($\geq$ 12 months under 18 months)
	WHO Paediatric Clinical Stage 3 or 4 ($\geq$ 12 months under 18 months)
Suspected ART drug failure	New WHO Clinical Stage 4 feature
	CD4 count at pre-treatment values or less
Implementing standardised ART reviews	Stable patients after 6 months
Managing first-line drug reactions	Managing lactic acidosis
	Managing skin reactions
Managing first-line regimen in children	Prescribing triomune-baby

Guideline	**Recommendation**
Prescribing CPT	Person with symptomatic HIV disease
	Child born to an HIV-positive woman
	Managing renal or hepatic toxicity in adults
	Managing severe haematological toxicity in adults
	Managing severe cutaneous reactions in children
	Managing renal or hepatic toxicity in children
	Dosing in younger children
Determining standard adult doses of ART drugs	Zidovudine
	Lamivudine
	Zalcitabine
	Emtricitabine
	Nelfinavir
	Atazanavir
Providing ART in special situations	Contraceptives and Neverapine
	Contraceptives and Efavirenz
	d4T/3TC/NVP not contraindicated in pregnancy
	Discontinuing NVP at the onset of labour
	Child born to HIV positive mother on ART
Managing ART patients who develop TB	Suspecting TB
	Managing patients who develop TB when patient is on ART
Managing Kaposi's sarcoma	Patient with mild to moderate disease
Managing drug toxicity	Diagnosing peripheral neuropathy
	Diagnosing pancreatitis
Managing symptoms during clinic visit	Managing new symptoms
Community IMCI	Managing child who can feed with convulsions
	Managing palmar pallor
	Managing swollen feet
	Managing child who can feed with severe red eyes
	managing child who can feed with red eyes and vision difficulties
	Managing child who can feed with chest indrawing

Appendix E

Representative sample of guideline recommendation change instances

Table E.1: Guideline recommendation changes stratified by type of CPG change

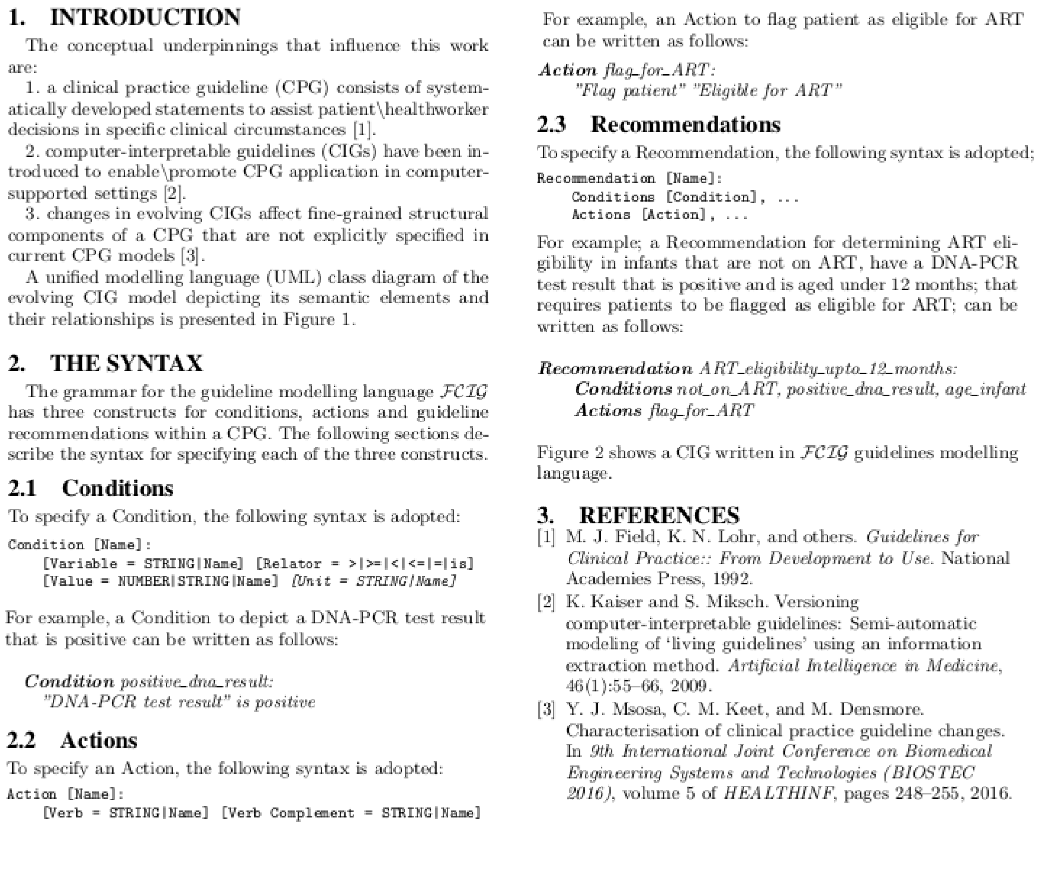
Change category	Guideline	Recommendation
Addition of a decision variable	Managingsuspected ART drug failure	Presenting with suspected ART drug failure
Change of a decision variable	Definition of ART eligibility	Adults – CD4 count below 250/mm3
	Definition of ART eligibility	Children over the age of 18 months - CD4 or TLC counts (<3 yrs)
	Definition of ART eligibility	Children over the age of 18 months - WHO Paediatric Clinical Stage 4 or WHO Paediatric Clinical Stage 3
Removal of a decision variable	Managing suspected ART drug failure	Suspected ART failure – new WHO stage 4
Change of a decision variable	Prescribing CPT	Dosages of CPT – Children aged 6 weeks to 5 months
	Prescribing CPT	Dosages of CPT – Children aged 5 – 14 years
	Standard Adult Doses of Antiretroviral Drugs	Lopinavir / ritonavir
Addition of a recommended action	Providing ART in special situations	ART in case of renal failure
Removal of a recommended action	Providing ART in special situations	ART in case of renal failure

Change category	**Guideline**	**Recommendation**
Change of an action verb complement	Determining standard adult doses of ART drugs	Stavudine
	Determining standard adult doses of ART drugs	Abacavir
Change of a recommended action	Managing suspected ART drug failure	Presenting with suspected ART drug failure
Addition of a recommendation	ART side–effects – stopping ART	Yellow eyes / hepatitis
	ART side-effects – stopping ART	Severe stomach pain and vomiting
	ART side-effects – stopping ART	Shortness of breath
	ART side-effects – stopping ART	Severe skin rash with blisters, involving eyes, mouth or genitals
	CD4 monitoring for ART eligibility	CD4 monitoring of patients in HIV Care Clinic followup Stopping CD4 monitoring
	Definition of ART eligibility	Presumed severe HIV disease (PSHD) Cryptococcal meningitis
	Monitoring of nutritional status	BMI under 17 – Non–pregnant adults 15 years and above
	Monitoring of nutritional status	MUAC less than 22cm
	Provider initiated family planning (PIFP)	Implementing routine PIFP in HIV clinic - women
	Provider initiated testing and counselling (PITC)	Tested negative more than 3 months ago
	Selecting regimen and formulation for continuation	Children on 1st line regimens when their weight is over 25kg
	Selecting regimen and formulation for continuation	If a woman became pregnant while on an ART regimen that contains EFV
Removal of a recommendation	Managing first-line drug reactions	Lactic acidosis/ Lipodystrophy syndrome
	Managing first-line drug reactions	Adverse reactions to first line regimen - child <3 yrs
	Managing first-line drug reactions	Child born to HIV positive mother on ART
	Managing first-line drug reactions	Initial phase of,anti-TB treatment - severely immuno-compromised patients
	Prescribing CPT	All patients eligible for ART
	Definition of ART eligibility	PSHD - recent HIV-related maternal death

Change category	**Guideline**	**Recommendation**
Removal of a recommendation	Definition of ART eligibility	At 18 months - If the test is positive - On ART
	Definition of ART eligibility	Children under the age of 18 months - CD4 or TLC counts below threshold values for starting ART (>= 12 months)
	Definition of ART eligibility	Children over the age of 18 months - CD4 or TLC counts (5 years,to <15 years)
	Definition of ART eligibility	Adults - WHO Clinical Stage 2 with a TLC <1200/mm3
	Managing Kaposi 's sarcoma	In patients with mild to moderate disease
	Managing Kaposi 's sarcoma	For others (severe)
	Determining priority for CD4 count testing	Patients in WHO stage 2
	Determining priority for CD4 count testing	Whenever ART failure is suspected
	Determining priority for CD4 count testing	Base-line
	Determining priority for CD4 count testing	Follow-up every 12 months
	Standard Adult Doses of Antiretroviral Drugs	Emtricitabine
	Managing symptoms during clinic visit	new or worsening symptoms since last visit

Appendix F

FCIG paper-based orientation



1. INTRODUCTION

The conceptual underpinnings that influence this work are:

1. a clinical practice guideline (CPG) consists of systematically developed statements to assist patient\healthworker decisions in specific clinical circumstances [1].

2. computer-interpretable guidelines (CIGs) have been introduced to enable\promote CPG application in computer-supported settings [2].

3. changes in evolving CIGs affect fine-grained structural components of a CPG that are not explicitly specified in current CPG models [3].

A unified modelling language (UML) class diagram of the evolving CIG model depicting its semantic elements and their relationships is presented in Figure 1.

2. THE SYNTAX

The grammar for the guideline modelling language $\mathcal{FCIG}$ has three constructs for conditions, actions and guideline recommendations within a CPG. The following sections describe the syntax for specifying each of the three constructs.

2.1 Conditions

To specify a Condition, the following syntax is adopted:

```
Condition [Name]:
    [Variable = STRING|Name] [Relator = >|>=|<|<=|=|is]
    [Value = NUMBER|STRING|Name] [Unit = STRING|Name]
```

For example, a Condition to depict a DNA-PCR test result that is positive can be written as follows:

Condition *positive_dna_result:*
"DNA-PCR test result" is positive

2.2 Actions

To specify an Action, the following syntax is adopted:

```
Action [Name]:
    [Verb = STRING|Name] [Verb Complement = STRING|Name]
```

For example, an Action to flag patient as eligible for ART can be written as follows:

Action *flag_for_ART:*
"Flag patient" "Eligible for ART"

2.3 Recommendations

To specify a Recommendation, the following syntax is adopted;

```
Recommendation [Name]:
    Conditions [Condition], ...
    Actions [Action], ...
```

For example; a Recommendation for determining ART eligibility in infants that are not on ART, have a DNA-PCR test result that is positive and is aged under 12 months; that requires patients to be flagged as eligible for ART; can be written as follows:

Recommendation *ART_eligibility_upto_12_months:*
Conditions *not_on_ART, positive_dna_result, age_infant*
Actions *flag_for_ART*

Figure 2 shows a CIG written in $\mathcal{FCIG}$ guidelines modelling language.

3. REFERENCES

[1] M. J. Field, K. N. Lohr, and others. *Guidelines for Clinical Practice:: From Development to Use.* National Academies Press, 1992.

[2] K. Kaiser and S. Miksch. Versioning computer-interpretable guidelines: Semi-automatic modeling of 'living guidelines' using an information extraction method. *Artificial Intelligence in Medicine*, 46(1):55–66, 2009.

[3] Y. J. Msosa, C. M. Keet, and M. Densmore. Characterisation of clinical practice guideline changes. In *9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016)*, volume 5 of *HEALTHINF*, pages 248–255, 2016.

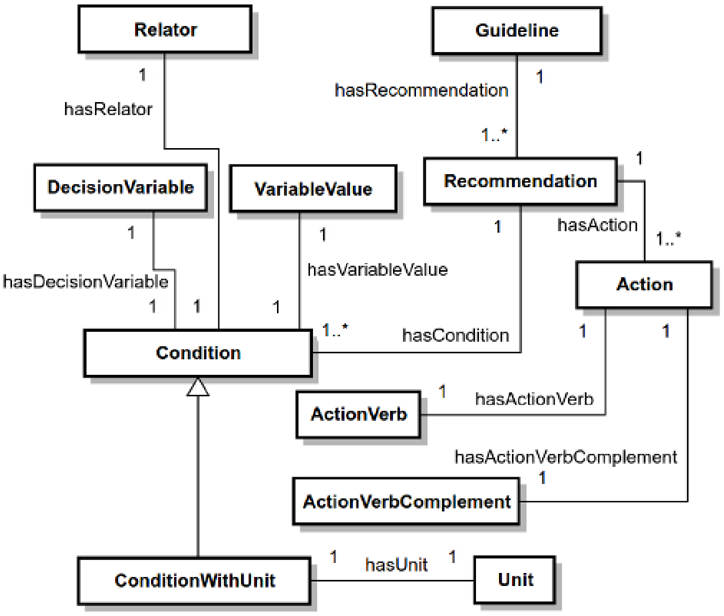
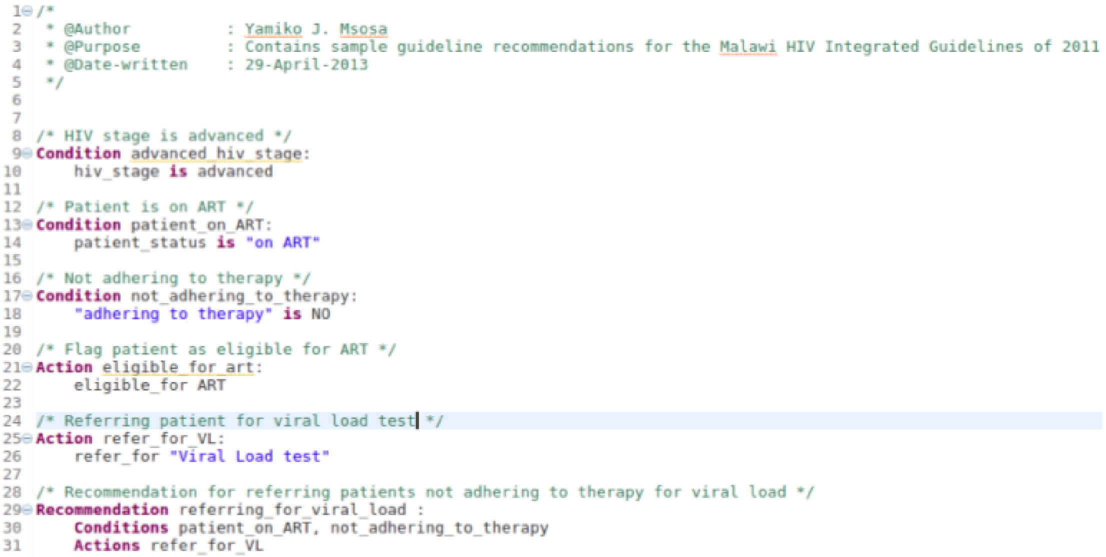


Figure 1: UML class diagram of a CIG metamodel



```
/*
 * @Author          : Yamiko J. Msosa
 * @Purpose         : Contains sample guideline recommendations for the Malawi HIV Integrated Guidelines of 2011
 * @Date-written    : 29-April-2013
 */


/* HIV stage is advanced */
Condition advanced_hiv_stage:
    hiv_stage is advanced

/* Patient is on ART */
Condition patient_on_ART:
    patient_status is "on ART"

/* Not adhering to therapy */
Condition not_adhering_to_therapy:
    "adhering to therapy" is NO

/* Flag patient as eligible for ART */
Action eligible_for_art:
    eligible_for ART

/* Referring patient for viral load test */
Action refer_for_VL:
    refer_for "Viral Load test"

/* Recommendation for referring patients not adhering to therapy for viral load */
Recommendation referring_for_viral_load :
    Conditions patient_on_ART, not_adhering_to_therapy
    Actions refer_for_VL
```

Figure 2: Sample CIG written in FCIG modelling language

Appendix G

Tasks for the CIG modelling experiments

G.1 Task one

ENCODING SUB-TASKS: Guideline recommendations for providing ART in special situations (2011)

1. For an HIV positive patient that is not on ART, is presenting with Anaemia and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 1A - d4T/3TC/NVP.
2. For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and is aged 6 weeks up to 14 years, flag patient as eligible for ART and prescribe Regimen 1P - d4T/3TC/NVP.
3. For an HIV positive patient that is not on ART, is presenting with Active TB and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.

G.2 Task two

UPDATE SUB-TASKS: Guideline recommendations for providing ART in special situations (2014)

1. For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 15 years or more~~ **whose weight is 35 kg or more**, flag

patient as eligible for ART and prescribe Regimen ~~1A – d4T/3TC/NVP~~ **5A – TDF/3TC/EFV**.

2. For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 6 weeks up to 14 years~~ **whose weight is less than 35 kg**, flag patient as eligible for ART and prescribe Regimen ~~1P – d4T/3TC/NVP~~ **0P – ABC/3TC+NVP**.

3. For an HIV positive patient that is not on ART, is presenting with Active TB and ~~is aged 15 years or more~~ **whose weight is 35 kg or more**, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.

4. For an HIV positive patient that is not on ART, is presenting with Active TB, **whose weight is less than 35 kg** and is aged ~~6 weeks to~~ under 3 years, flag patient as eligible for ART and prescribe Regimen 2P – AZT/3TC/NVP.

5. For an HIV positive patient that is not on ART, is presenting with Active TB, **whose weight is less than 35 kg** and is aged from 3 years ~~up to 14 years~~, flag patient as eligible for ART and prescribe Regimen 4P – AZT/3TC+EFV.

6. For an HIV positive patient that is not on ART, is presenting with Jaundice and ~~is aged 6 weeks up to 14 years~~ **whose weight is less than 35 kg**, flag patient as eligible for ART, prescribe Regimen 4P – AZT/3TC+EFV and Refer to District or Central Hospital (Secondary care).

7. For an HIV positive patient that is not on ART, is presenting with Jaundice and ~~is aged 15 years or more~~ **whose weight is 35 kg or more**, flag patient as eligible for ART, prescribe Regimen ~~3A – d4T/3TC+EFV~~ **5A – TDF/3TC/EFV** and Refer to District or Central Hospital (Secondary care).

8. For an HIV positive patient that is not on ART and is pregnant ~~and are in their 2nd trimester of pregnancy~~, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.

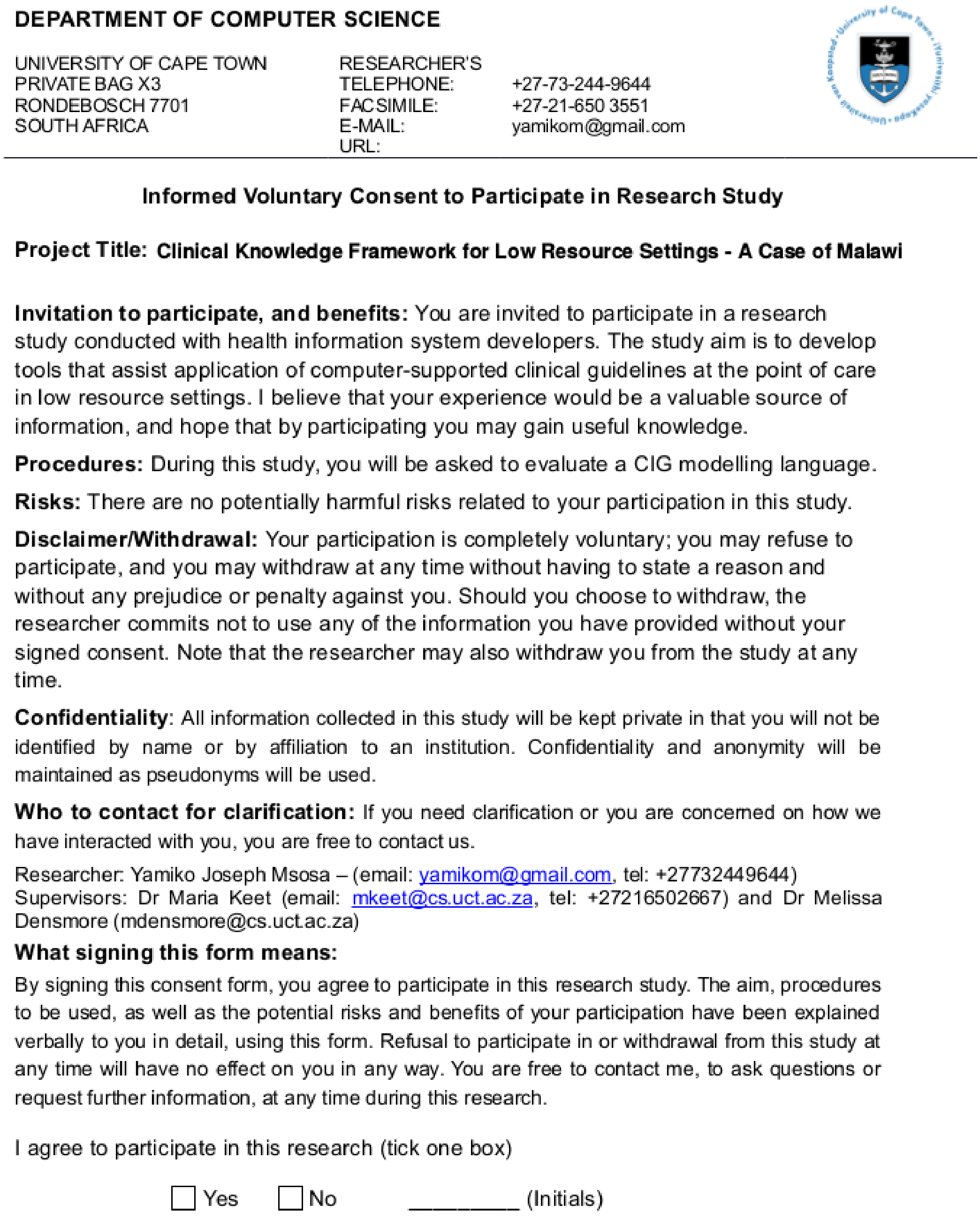
9. For an HIV positive patient that is not on ART, is presenting with renal failure ~~and is aged 6 weeks up to 14 years~~, flag patient as eligible for ART, prescribe Regimen ~~2P – AZT/3TC/NVP~~ **0 – ABC/3TC+NVP** and Refer to District or Central Hospital (Secondary care).

10. ~~For an HIV positive patient that is not on ART, is presenting with renal failure and is aged 15 years or more, flag patient as eligible for ART, prescribe Regimen 4A – AZT/3TC + EFV and Refer to District or Central Hospital (Secondary care).~~

Appendix H

Documentation for experiments

H.1 Consent form



DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF CAPE TOWN
PRIVATE BAG X3
RONDEBOSCH 7701
SOUTH AFRICA

RESEARCHER'S
TELEPHONE: +27-73-244-9644
FACSIMILE: +27-21-650 3551
E-MAIL: yamikom@gmail.com
URL:

Informed Voluntary Consent to Participate in Research Study

Project Title: Clinical Knowledge Framework for Low Resource Settings - A Case of Malawi

Invitation to participate, and benefits: You are invited to participate in a research study conducted with health information system developers. The study aim is to develop tools that assist application of computer-supported clinical guidelines at the point of care in low resource settings. I believe that your experience would be a valuable source of information, and hope that by participating you may gain useful knowledge.

Procedures: During this study, you will be asked to evaluate a CIG modelling language.

Risks: There are no potentially harmful risks related to your participation in this study.

Disclaimer/Withdrawal: Your participation is completely voluntary; you may refuse to participate, and you may withdraw at any time without having to state a reason and without any prejudice or penalty against you. Should you choose to withdraw, the researcher commits not to use any of the information you have provided without your signed consent. Note that the researcher may also withdraw you from the study at any time.

Confidentiality: All information collected in this study will be kept private in that you will not be identified by name or by affiliation to an institution. Confidentiality and anonymity will be maintained as pseudonyms will be used.

Who to contact for clarification: If you need clarification or you are concerned on how we have interacted with you, you are free to contact us.

Researcher: Yamiko Joseph Msosa – (email: yamikom@gmail.com, tel: +27732449644)
Supervisors: Dr Maria Keet (email: mkeet@cs.uct.ac.za, tel: +27216502667) and Dr Melissa Densmore (mdensmore@cs.uct.ac.za)

What signing this form means:

By signing this consent form, you agree to participate in this research study. The aim, procedures to be used, as well as the potential risks and benefits of your participation have been explained verbally to you in detail, using this form. Refusal to participate in or withdrawal from this study at any time will have no effect on you in any way. You are free to contact me, to ask questions or request further information, at any time during this research.

I agree to participate in this research (tick one box)

□ Yes □ No _________ (Initials)

H.2 Experiment procedure explanation

We are comparing two CIG modelling languages, Language A and Language B. You will be randomly given the modelling basics of one of the two languages via a self-paced tutorial through a wiki. Thereafter, you will perform two CIG modelling tasks using the language assigned to you. You will then evaluate the CIG modelling language that you used. You will take a five minute break, after which, you will be given the basics of the other modelling language via a wiki. Thereafter, you will perform the same two CIG modelling tasks using the second language. You will then evaluate the second CIG modelling language as well.

H.3 Group one experiment instructions

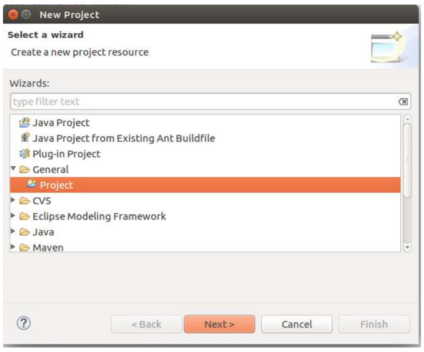
H.3.1 Condition A instructions

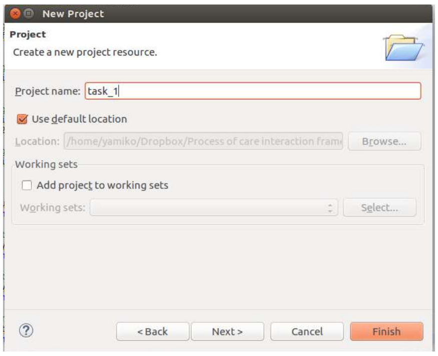
<u>Task 1 – Create Guideline Recommendations for Providing Antiretroviral Therapy (ART) in Special Situations</u>

1. Open Eclipse IDE using the default workspace

2. Switch to a new workspace named '**/first_experiment_part_1/**' by using the following menu: **'File > Switch Workspace > Other'** and typing **/first_experiment_part_1/**

3. Create a new general project named 'task_1' by using the following menu '**File > New > Project...**' and select project of type general as shown below:





Project name: task_1

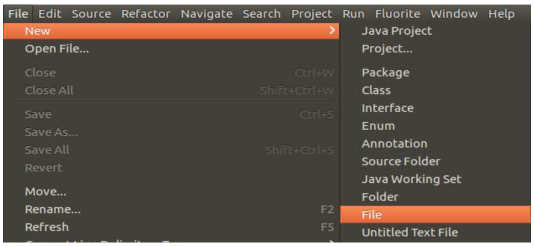
☑ Use default location

Location: /home/yamiko/Dropbox/Process of care interaction frame

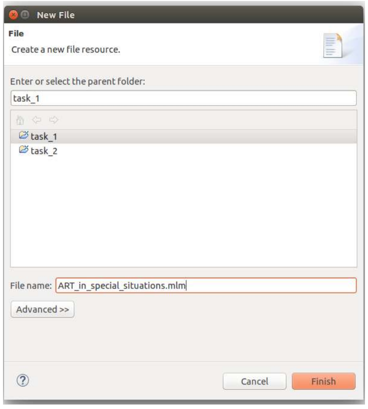
□ Add project to working sets

4. Click finish to create the new project

5. Create a new file named 'ART_in_special_situations.mlm' using the 'File > New > File' menu as shown below:



6. Type in the file name and click finish as shown below:



File

Create a new file resource.

Enter or select the parent folder:

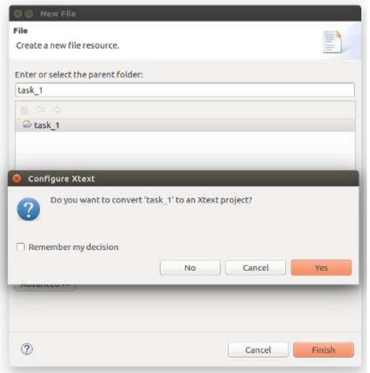
task_1

task_1

task_2

File name: ART_in_special_situations.mlm

7. Click ‘Yes’ on the dialogue box similar to the one shown below:



File

Create a new file resource.

Enter or select the parent folder:

task_1

task_1

Do you want to convert 'task_1' to an Xtext project?

□ Remember my decision

8. For each itemised **encoding sub-task** below (i, ii, iii, ...) :

8.1. Record your start time on the self reporting form in the corresponding field

8.2. Encode the guideline recommendation accordingly using ***Language A***.

8.3. Record the stop time at the end or when you are unable to complete the task

ENCODING SUB-TASKS: Guideline recommendations for providing ART in special situations (2011)

i. For an HIV positive patient that is not on ART, is presenting with Anaemia and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 1A - d4T/3TC/NVP.

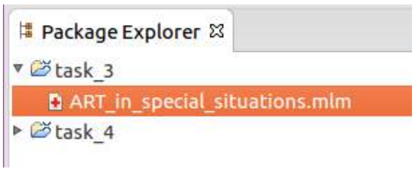
ii. For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and is aged 6 weeks up to 14 years, flag patient as eligible for ART and prescribe Regimen 1P - d4T/3TC/NVP.

iii. For an HIV positive patient that is not on ART, is presenting with Active TB and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.

Task 2 – Update Guideline Recommendations for Providing ART in Special Situations

1. Switch to a new workspace named '**/first_experiment_part_2/**' by using the following menu: **'File > Switch Workspace > Other'** and typing **/first_experiment_part_2/**

2. Select the project named 'task_3' and double click 'ART_in_special_situations.mlm' in Package Explorer as shown below:



3. For each itemised **update sub-task** below (i, ii, iii, ...) :

   3.1. Record your start time on the self reporting form in the corresponding field

   3.2. Update the guideline recommendation accordingly using *Language A*.

   3.3. Record the stop time at the end or when you are unable to complete the task

UPDATE SUB-TASKS: Guideline recommendations for providing ART in special situations (2014)

i. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen ~~1A – d4T/3TC/NVP~~* ***5A – TDF/3TC/EFV.***

ii. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 6 weeks up to 14 years~~* ***whose weight is less than 35 kg****, flag patient as eligible for ART and prescribe Regimen ~~1P – d4T/3TC/NVP~~* ***0P – ABC/3TC+NVP****.*

iii. *For an HIV positive patient that is not on ART, is presenting with Active TB and ~~is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

iv. *For an HIV positive patient that is not on ART, is presenting with Active TB, **whose weight is less than 35 kg** and is aged ~~6 weeks to~~ under 3 years, flag patient as eligible for ART and prescribe Regimen 2P – AZT/3TC/NVP.*

v. *For an HIV positive patient that is not on ART, is presenting with Active TB, **whose weight is less than 35 kg** and is aged from 3 years ~~up to 14 years~~, flag patient as eligible for ART and prescribe Regimen 4P – AZT/3TC+EFV.*

vi. *For an HIV positive patient that is not on ART, is presenting with Jaundice and ~~is aged 6 weeks up to 14 years~~ **whose weight is less than 35 kg**, flag patient as eligible for ART, prescribe Regimen 4P – AZT/3TC+EFV and Refer to District or Central Hospital (Secondary care).*

vii. *For an HIV positive patient that is not on ART, is presenting with Jaundice and ~~is aged 15 years or more~~ **whose weight is 35 kg or more**, flag patient as eligible for ART, prescribe Regimen ~~3A – d4T/3TC+EFV~~ **5A – TDF/3TC/EFV** and Refer to District or Central Hospital (Secondary care).*

viii. *For an HIV positive patient that is not on ART and is pregnant ~~and are in their 2nd trimester of pregnancy~~, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

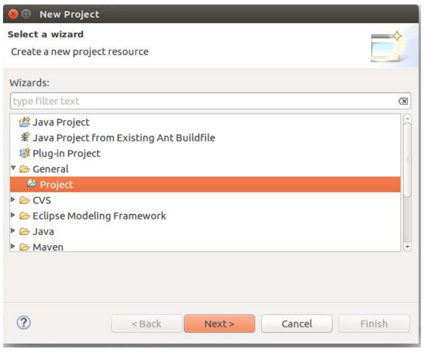
ix. *For an HIV positive patient that is not on ART, is presenting with renal failure ~~and is aged 6 weeks up to 14 years~~, flag patient as eligible for ART, prescribe Regimen ~~2P – AZT/3TC/NVP~~ **0 – ABC/3TC+NVP** and Refer to District or Central Hospital (Secondary care).*

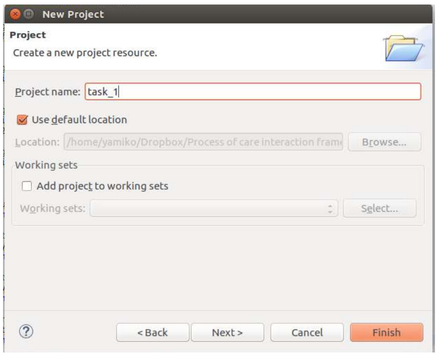
~~x. For an HIV positive patient that is not on ART, is presenting with renal failure and is aged 15 years or more, flag patient as eligible for ART, prescribe Regimen 4A – AZT/3TC + EFV and Refer to District or Central Hospital (Secondary care).~~

H.3.2 Condition B instructions

Task 1 – Create Guideline Recommendations for Providing Antiretroviral Therapy (ART) in Special Situations

1. Open Eclipse IDE using the default workspace
2. Switch to a new workspace named '**/second_experiment_part_1/**' by using the following menu: **'File > Switch Workspace > Other'** and typing **/second_experiment_part_1/**
3. Create a new general project named 'task_1' by using the following menu '**File > New > Project...**' and select project of type general as shown below:



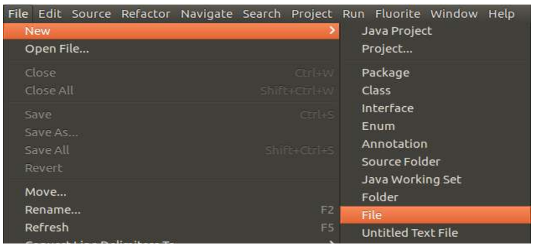


Project name: task_1

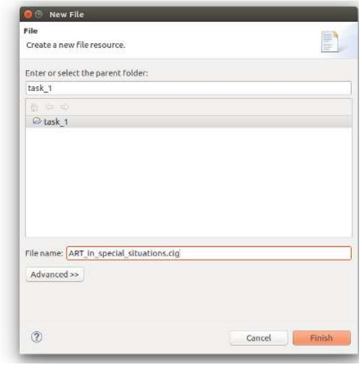
☑ Use default location

Location: /home/yamiko/Dropbox/Process of care interaction frame

4. Click finish to create the new project
5. Create a new file named 'ART_in_special_situations.cig' using the 'File > New > File' menu as shown below:



6. Type in the file name and click finish as shown below:



Create a new file resource.

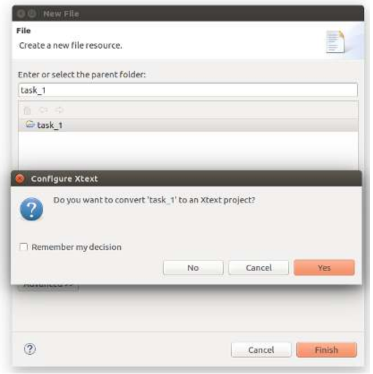
Enter or select the parent folder:

task_1

task_1

File name: ART_in_special_situations.cig

7. Click ‘Yes’ on the dialogue box similar to the one shown below:



Create a new file resource.

Enter or select the parent folder:

task_1

task_1

Do you want to convert 'task_1' to an Xtext project?

□ Remember my decision

8. For each itemised **encoding sub-task** below (i, ii, iii, ...) :

   8.1. Record your start time on the self reporting form in the corresponding field

   8.2. Encode the guideline recommendation accordingly using ***Language B***.

   8.3. Record the stop time at the end or when you are unable to complete the task

ENCODING SUB-TASKS: Guideline recommendations for providing ART in special situations (2011)

i. *For an HIV positive patient that is not on ART, is presenting with Anaemia and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 1A - d4T/3TC/NVP.*

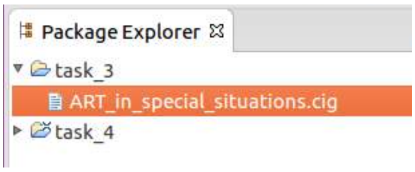
ii. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and is aged 6 weeks up to 14 years, flag patient as eligible for ART and prescribe Regimen 1P - d4T/3TC/NVP.*

iii. *For an HIV positive patient that is not on ART, is presenting with Active TB and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

Task 2 – Update Guideline Recommendations for Providing ART in Special Situations

1. Switch to a new workspace named '**/second_experiment_part_2/**' by using the following menu: **'File > Switch Workspace > Other'** and typing **/second_experiment_part_2/**

2. Select the project named 'task_3' and double click 'ART_in_special_situations.cig' in Package Explorer as shown below:



3. For each itemised **update sub-task** below (i, ii, iii, ...) :

   3.1. Record your start time on the self reporting form in the corresponding field

   3.2. Update the guideline recommendation accordingly using ***Language B***.

   3.3. Record the stop time at the end or when you are unable to complete the task

UPDATE SUB-TASKS: Guideline recommendations for providing ART in special situations (2014)

i. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen ~~1A – d4T/3TC/NVP~~* ***5A – TDF/3TC/EFV.***

ii. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 6 weeks up to 14 years~~* ***whose weight is less than 35 kg****, flag patient as eligible for ART and prescribe Regimen ~~1P – d4T/3TC/NVP~~* ***0P – ABC/3TC+NVP****.*

iii. *For an HIV positive patient that is not on ART, is presenting with Active TB and ~~is aged 15 years or more~~ **whose weight is 35 kg or more**, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

iv. *For an HIV positive patient that is not on ART, is presenting with Active TB, **whose weight is less than 35 kg** and is aged ~~6 weeks to~~ under 3 years, flag patient as eligible for ART and prescribe Regimen 2P – AZT/3TC/NVP.*

v. *For an HIV positive patient that is not on ART, is presenting with Active TB, **whose weight is less than 35 kg** and is aged from 3 years ~~up to 14 years~~, flag patient as eligible for ART and prescribe Regimen 4P – AZT/3TC+EFV.*

vi. *For an HIV positive patient that is not on ART, is presenting with Jaundice and ~~is aged 6 weeks up to 14 years~~ **whose weight is less than 35 kg**, flag patient as eligible for ART, prescribe Regimen 4P – AZT/3TC+EFV and Refer to District or Central Hospital (Secondary care).*

vii. *For an HIV positive patient that is not on ART, is presenting with Jaundice and ~~is aged 15 years or more~~ **whose weight is 35 kg or more**, flag patient as eligible for ART, prescribe Regimen ~~3A – d4T/3TC+EFV~~ **5A – TDF/3TC/EFV** and Refer to District or Central Hospital (Secondary care).*

viii. *For an HIV positive patient that is not on ART and is pregnant ~~and are in their 2nd trimester of pregnancy~~, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

ix. *For an HIV positive patient that is not on ART, is presenting with renal failure ~~and is aged 6 weeks up to 14 years~~, flag patient as eligible for ART, prescribe Regimen ~~2P – AZT/3TC/NVP~~ **0 – ABC/3TC+NVP** and Refer to District or Central Hospital (Secondary care).*

x. ~~*For an HIV positive patient that is not on ART, is presenting with renal failure and is aged 15 years or more, flag patient as eligible for ART, prescribe Regimen 4A – AZT/3TC + EFV and Refer to District or Central Hospital (Secondary care).*~~

H.4 Group two experiment instructions

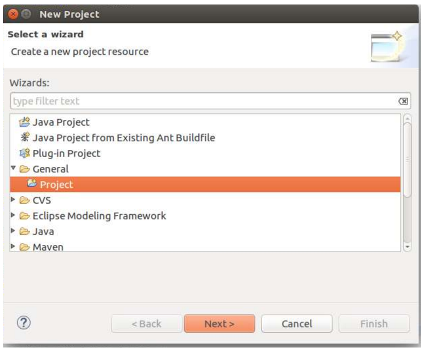
H.4.1 Condition B instructions

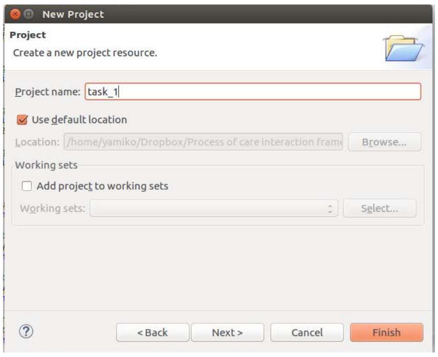
Task 1 – Create Guideline Recommendations for Providing Antiretroviral Therapy (ART) in Special Situations

1. Open Eclipse IDE using the default workspace

2. Switch to a new workspace named '**/FirstExperimentPart1/**' by using the following menu: **'File > Switch Workspace > Other'** and typing **/FirstExperimentPart1/**

3. Create a new general project named 'task_1' by using the following menu '**File > New > Project...**' and select project of type general as shown below:





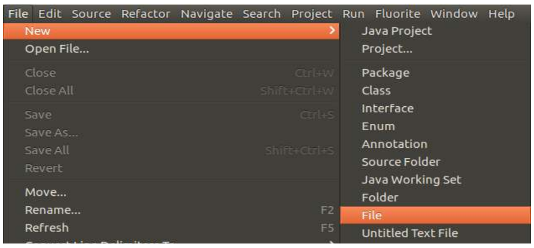
Project name: task_1

☑ Use default location

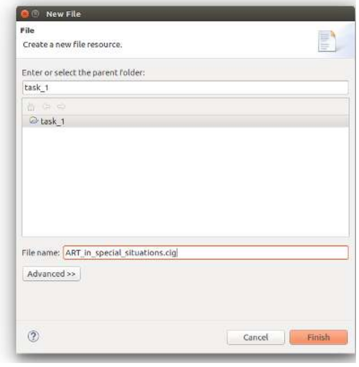
Location: /home/yamiko/Dropbox/Process of care interaction fram

4. Click finish to create the new project

5. Create a new file named 'ART_in_special_situations.cig' using the 'File > New > File' menu as shown below:



6. Type in the file name and click finish as shown below:



New File

File

Create a new file resource.

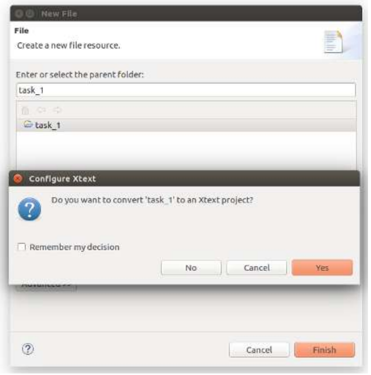
Enter or select the parent folder:

task_1

task_1

File name: ART_in_special_situations.cig

7. Click ‘Yes’ on the dialogue box similar to the one shown below:



New File

File

Create a new file resource.

Enter or select the parent folder:

task_1

task_1

Configure Xtext

Do you want to convert 'task_1' to an Xtext project?

□ Remember my decision

8. For each itemised **encoding sub-task** below (i, ii, iii, ...) :

    8.1. Record your start time on the self reporting form in the corresponding field

    8.2. Encode the guideline recommendation accordingly using ***Language A***.

    8.3. Record the stop time at the end or when you are unable to complete the task

ENCODING SUB-TASKS: Guideline recommendations for providing ART in special situations (2011)

i. For an HIV positive patient that is not on ART, is presenting with Anaemia and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 1A - d4T/3TC/NVP.

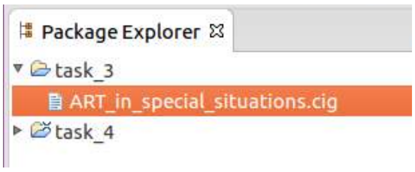
ii. For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and is aged 6 weeks up to 14 years, flag patient as eligible for ART and prescribe Regimen 1P - d4T/3TC/NVP.

iii. For an HIV positive patient that is not on ART, is presenting with Active TB and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.

Task 2 – Update Guideline Recommendations for Providing ART in Special Situations

1. Switch to a new workspace named '**/FirstExperimentPart2/**' by using the following menu: **'File > Switch Workspace > Other'** and typing **/FirstExperimentPart2/**

2. Select the project named 'task_3' and double click 'ART_in_special_situations.cig' in Package Explorer as shown below:



3. For each itemised **update sub-task** below (i, ii, iii, ...) :

   3.1. Record your start time on the self reporting form in the corresponding field

   3.2. Update the guideline recommendation accordingly using ***Language A***.

   3.3. Record the stop time at the end or when you are unable to complete the task

UPDATE SUB-TASKS: Guideline recommendations for providing ART in special situations (2014)

i. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen ~~1A – d4T/3TC/NVP~~* ***5A – TDF/3TC/EFV.***

ii. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 6 weeks up to 14 years~~* ***whose weight is less than 35 kg****, flag patient as eligible for ART and prescribe Regimen ~~1P – d4T/3TC/NVP~~* ***0P – ABC/3TC+NVP****.*

iii. *For an HIV positive patient that is not on ART, is presenting with Active TB and~~ is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

iv. For an HIV positive patient that is not on ART, is presenting with Active TB, ***whose weight is less than 35 kg*** *and is aged ~~6 weeks to~~ under 3 years, flag patient as eligible for ART and prescribe Regimen 2P – AZT/3TC/NVP.*

v. For an HIV positive patient that is not on ART, is presenting with Active TB, ***whose weight is less than 35 kg*** *and is aged from 3 years~~ up to 14 years~~, flag patient as eligible for ART and prescribe Regimen 4P – AZT/3TC+EFV.*

vi. For an HIV positive patient that is not on ART, is presenting with Jaundice and~~ is aged 6 weeks up to 14 years~~ ***whose weight is less than 35 kg****, flag patient as eligible for ART, prescribe Regimen 4P – AZT/3TC+EFV and Refer to District or Central Hospital (Secondary care).*

vii. For an HIV positive patient that is not on ART, is presenting with Jaundice and~~ is aged 15 years or more~~ ***whose weight is 35 kg or more****, flag patient as eligible for ART, prescribe Regimen ~~3A – d4T/3TC+EFV~~* ***5A – TDF/3TC/EFV*** *and Refer to District or Central Hospital (Secondary care).*

viii. For an HIV positive patient that is not on ART and is pregnant ~~and are in their 2nd trimester of pregnancy~~, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.

ix. For an HIV positive patient that is not on ART, is presenting with renal failure ~~and is aged 6 weeks up to 14 years~~, flag patient as eligible for ART, prescribe Regimen ~~2P – AZT/3TC/NVP~~ ***0 – ABC/3TC+NVP*** *and Refer to District or Central Hospital (Secondary care).*

x. ~~For an HIV positive patient that is not on ART, is presenting with renal failure and is aged 15 years or more, flag patient as eligible for ART, prescribe Regimen 4A – AZT/3TC + EFV and Refer to District or Central Hospital (Secondary care).~~

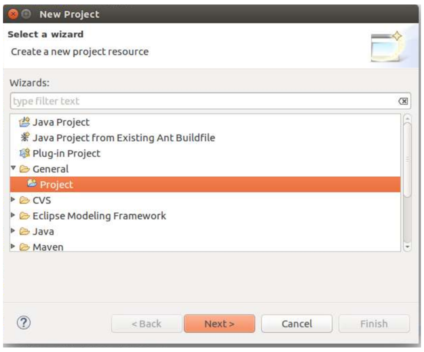
H.4.2 Condition A instructions

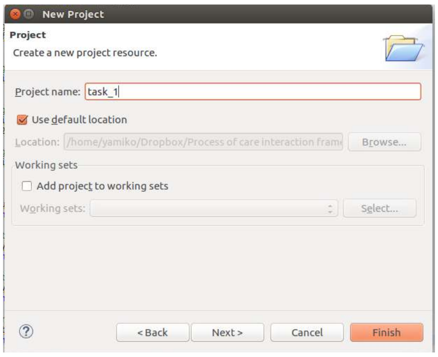
Task 1 – Create Guideline Recommendations for Providing Antiretroviral Therapy (ART) in Special Situations

1. Open Eclipse IDE using the default workspace

2. Switch to a new workspace named ‘**SecondExperimentPart1/**’ by using the following menu: **‘File > Switch Workspace > Other’** and typing **/SecondExperimentPart1/**

3. Create a new general project named ‘task_1’ by using the following menu ‘**File > New > Project...**’ and select project of type general as shown below:





Project name: task_1

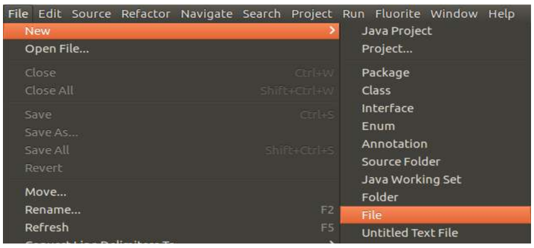
☑ Use default location

Location: /home/yamiko/Dropbox/Process of care interaction frame

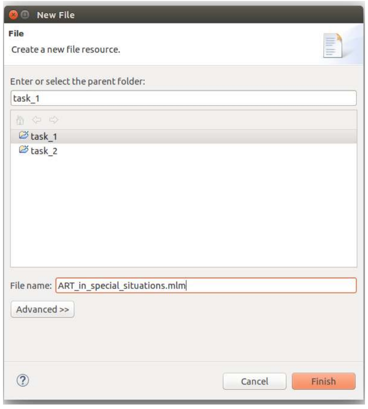
☐ Add project to working sets

4. Click finish to create the new project

5. Create a new file named ‘ART_in_special_situations.mlm’ using the ‘File > New > File’ menu as shown below:



6. Type in the file name and click finish as shown below:



New File

File

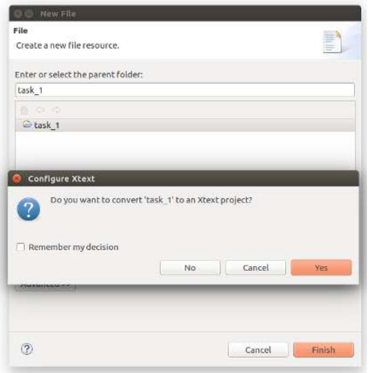
Create a new file resource.

Enter or select the parent folder: task_1

- task_1
- task_2

File name: ART_in_special_situations.mlm

7. Click ‘Yes’ on the dialogue box similar to the one shown below:



New File

File

Create a new file resource.

Enter or select the parent folder: task_1

- task_1

Configure Xtext

Do you want to convert 'task_1' to an Xtext project?

□ Remember my decision

8. For each itemised **encoding sub-task** below (i, ii, iii, ...) :

   8.1. Record your start time on the self reporting form in the corresponding field

   8.2. Encode the guideline recommendation accordingly using ***Language B***.

   8.3. Record the stop time at the end or when you are unable to complete the task

ENCODING SUB-TASKS: Guideline recommendations for providing ART in special situations (2011)

i. *For an HIV positive patient that is not on ART, is presenting with Anaemia and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 1A - d4T/3TC/NVP.*

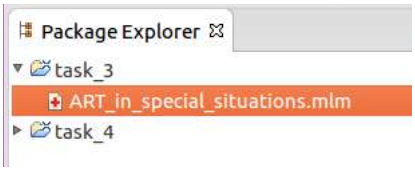
ii. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and is aged 6 weeks up to 14 years, flag patient as eligible for ART and prescribe Regimen 1P - d4T/3TC/NVP.*

iii. *For an HIV positive patient that is not on ART, is presenting with Active TB and is aged 15 years or more, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

Task 2 – Update Guideline Recommendations for Providing ART in Special Situations

1. Switch to a new workspace named ‘**/SecondExperimentPart2/**’ by using the following menu: **‘File > Switch Workspace > Other’** and typing **/SecondExperimentPart2/**

2. Select the project named ‘task_3’ and double click ‘ART_in_special_situations.mlm’ in Package Explorer as shown below:



3. For each itemised **update sub-task** below (i, ii, iii, ...) :

   3.1. Record your start time on the self reporting form in the corresponding field

   3.2. Update the guideline recommendation accordingly using ***Language B***.

   3.3. Record the stop time at the end or when you are unable to complete the task

UPDATE SUB-TASKS: Guideline recommendations for providing ART in special situations (2014)

i. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen ~~1A – d4T/3TC/NVP~~* ***5A – TDF/3TC/EFV.***

ii. *For an HIV positive patient that is not on ART, is presenting with Anaemia (<8g/dl) and ~~is aged 6 weeks up to 14 years~~* ***whose weight is less than 35 kg****, flag patient as eligible for ART and prescribe Regimen ~~1P – d4T/3TC/NVP~~* ***0P – ABC/3TC+NVP****.*

iii. *For an HIV positive patient that is not on ART, is presenting with Active TB and ~~is aged 15 years or more~~* ***whose weight is 35 kg or more****, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

iv. *For an HIV positive patient that is not on ART, is presenting with Active TB,* ***whose weight is less than 35 kg*** *and is aged* ~~*6 weeks to*~~ *under 3 years, flag patient as eligible for ART and prescribe Regimen 2P – AZT/3TC/NVP.*

v. *For an HIV positive patient that is not on ART, is presenting with Active TB,* ***whose weight is less than 35 kg*** *and is aged from 3 years*~~*up to 14 years*~~*, flag patient as eligible for ART and prescribe Regimen 4P – AZT/3TC+EFV.*

vi. *For an HIV positive patient that is not on ART, is presenting with Jaundice and*~~*is aged 6 weeks up to 14 years*~~ ***whose weight is less than 35 kg****, flag patient as eligible for ART, prescribe Regimen 4P – AZT/3TC+EFV and Refer to District or Central Hospital (Secondary care).*

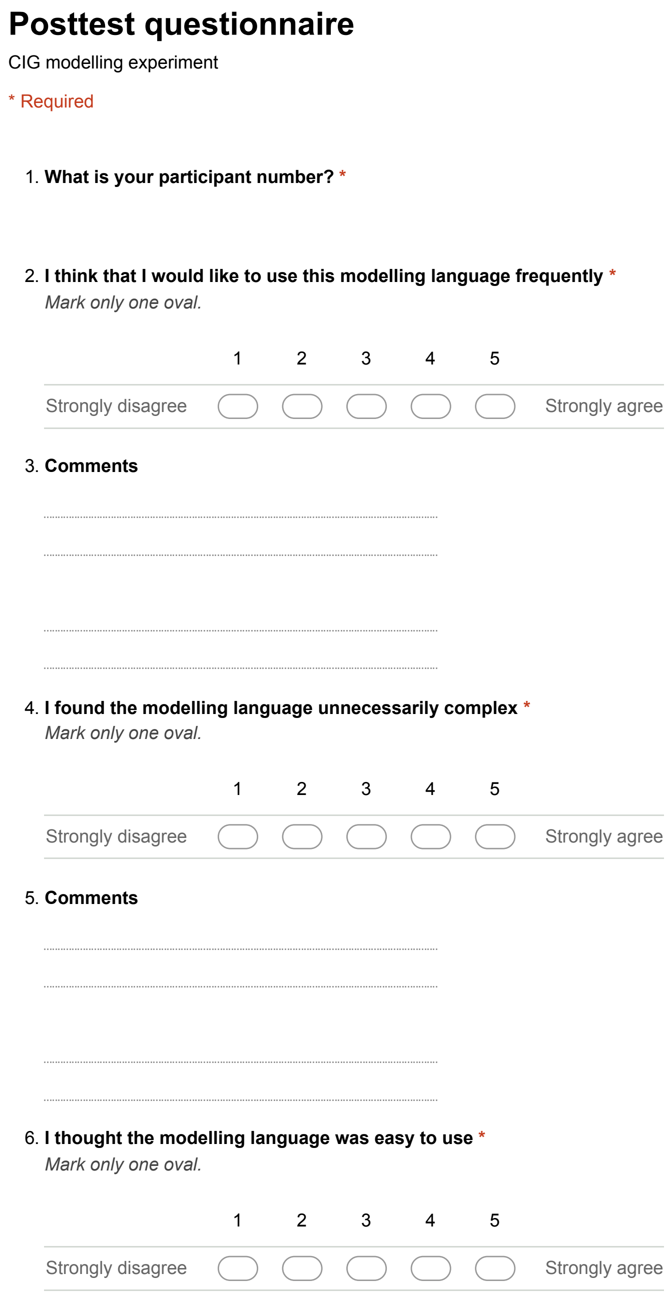
vii. *For an HIV positive patient that is not on ART, is presenting with Jaundice and*~~*is aged 15 years or more*~~ ***whose weight is 35 kg or more****, flag patient as eligible for ART, prescribe Regimen* ~~*3A – d4T/3TC+EFV*~~ ***5A – TDF/3TC/EFV*** *and Refer to District or Central Hospital (Secondary care).*

viii. *For an HIV positive patient that is not on ART and is pregnant* ~~*and are in their 2nd trimester of pregnancy*~~*, flag patient as eligible for ART and prescribe Regimen 5A – TDF/3TC/EFV.*

ix. *For an HIV positive patient that is not on ART, is presenting with renal failure* ~~*and is aged 6 weeks up to 14 years*~~*, flag patient as eligible for ART, prescribe Regimen* ~~*2P – AZT/3TC/NVP*~~ ***0 – ABC/3TC+NVP*** *and Refer to District or Central Hospital (Secondary care).*

~~*x. For an HIV positive patient that is not on ART, is presenting with renal failure and is aged 15 years or more, flag patient as eligible for ART, prescribe Regimen 4A – AZT/3TC + EFV and Refer to District or Central Hospital (Secondary care).*~~

H.5 Experiment posttest questionnaire



Posttest questionnaire

CIG modelling experiment

* Required

1. **What is your participant number? ***

2. **I think that I would like to use this modelling language frequently ***
   Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

3. **Comments**

4. **I found the modelling language unnecessarily complex ***
   Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

5. **Comments**

6. **I thought the modelling language was easy to use ***
   Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

7. **Comments**

8. **I think that I would need the support of a technical person to be able to use this modelling language** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

9. **Comments**

10. **I found the various functions in this modelling language were well integrated** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

11. **Comments**

12. **I thought there was too much inconsistency in this modelling language** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

13. **Comments**

14. **I would imagine that most people would learn to use this modelling language very quickly** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

15. **Comments**

16. **I found the modelling language very cumbersome to use** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

17. **Comments**

18. **I felt very confident using the modelling language** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

19. **Comments**

20. **I needed to learn a lot of things before I could get going with this modelling language** *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

21. **Comments**

22. **Which modelling features, if any, did you find useful whilst modelling the clinical guidelines?**

23. **Are there any modelling language features that are missing to support modelling of clinical guidelines?**

Appendix I

Contextual inquiry interview notes

I.1 Developer semi-structured interview notes

I.2 Health worker semi-structured interview notes

Table I.1: Developer interview responses - part A

.

Partici-pant	**Can computers improve H/C delivery?**	**How?**	**Challenges faced when developing e-Health solutions**	**Imple-mented CPGs in an EMR system before?**	**How?**
d1	Yes	Speed up work, Medical history lookup, medical research	Data gaps	Yes	Hard coded into the EMR
d2	Yes (with well trained medical and IT practitioners)	Record keeping, diagnosis support, robotics, surgery/precision	Unrealistic demands from the end-users, unknown requirements	Yes	Hard coded IF ... THEN ... ELSE ... rules
d3	Yes	Decision support, minimising errors, keeping past medical history	Knowledge transfer to the users	Yes	Reminders and also depends on the situation at hand
d4	Yes	Data systems and monitoring	Focus is on the data than the patient, No way of introducing systems effortlessly	Yes	Spec was straight forward so the rules were hard-coded

Table I.2: Developer interview responses - part A (continued)

.

Partici-pant	**Can computers improve H/C delivery?**	**How?**	**Challenges faced when developing e-Health solutions**	**Imple-mented CPGs in an EMR system before?**	**How?**
d5	Yes	Creating tools that complement what health workers do	Getting the users to specify what they actually need	Yes	Demonstrated by hard-coding
d6	Yes, when used in the right context, for the right purpose	Record keeping, decision support, M & E reports, drug stocks management, automation for efficiency	Lack of clear vision for e-Health solutions, conflicting stakeholder interests, lack of technical know-how on effective and efficient technical solution implementation	Yes	Encoded in a web application, as alerts
d7	Yes, when they complement the health worker	Assisting health workers do their job e.g. calculating BMI	Building systems that match an individual clinic's workflow	Yes	Hard-coding guidelines/building EMR functionality around guidelines
d8	Yes	collecting, storing, transmitting & presenting medical data; automating CPGs/decision support; data analysis & reporting	Poor connectivity, high work load on users	Yes	Hard -coded/Integral part of business logic of EMR
d9	Yes	Medical history lookup, improving access, decision support	Different user definitions/specifications of the same problem, unrealistic deadlines by product owners, redefinition of application features at any point of the development phase	Yes	Hard-coded in the EMR

TABLE I.3: Developer interview responses - Part B

Partici-pant	**What challenges are you facing when implementing CPGs in an EMR?**	**What would be the ideal way?**
d1	Separation of data from logic	Separated from other layers
d2	When guidelines change, one needs to search where to make corresponding changes in the code	Separate rules from logic/servlet/controller
d3	Keeping up with changes in the guideline. Sources of information are not formal.	Kind of tricky. Depends on the architecture
d4	Presentation of the guidelines	Guidelines change, separate guidelines according to areas
d5	Sometimes users do not provide the guidelines	Separate guidelines from other layers of the software. Enable end-user to change guideline as opposed to the developer. Consider practicality of implementing CPGs
d6	Keeping the systems updated with latest guidelines	through CPG engines that execute guidelines in a standardised form
d7	Adapting guideline -based systems to a clinic specific workflow	Make sure implementation is flexible(not needing an expert to modify) in the system and allow for health worker to override recommendations
d8	Changes in guidelines requires deep understanding of the system implementation; It is hard maintain guidelines that apply across multiple EMRs or versions of an EMR	Using some type of rule engine that is integrated with the EMR
d9	Guideline logic is tedious to maintain	In a way that they can be easily used in all EMR applications/software

Table I.4: Health worker interview responses - part A

Participant	**Location**	**Location classification**	**Age (years)**	**Gender**	**Highest qualification**	**Year qualification was obtained**
h1	Area 18	Peri-urban	33	F	Certificate in Clinical Medicine	2009
h2	Area 18	Peri-urban	43	M	Certificate in Clinical Medicine	2001
h3	KCH	Tertiary Hospital	42	M	Internal Medicine Fellowship (Consultant)	2012
h4	KCH	Tertiary Hospital	27	F	Six year Medical degree (MBBS)	2009
h5	KCH	Tertiary Hospital	27	M	Six year Medical degree (MBBS)	2009
h6	Chileka	Rural	33	M	Certificate in Clinical Medicine	2008
h7	Lumbadzi	Rural	25	M	BSc Nursing and Midwifery	2011
h8	Lumbadzi	Rural	33	F	Diploma in Nursing	2009
h9	Lumbadzi	Rural	28	F	Certificate in Clinical Medicine	2008
h10	Mtenthera	Rural	34	M	Certificate in Clinical Medicine	2003
h11	Mbabvi	Rural	32	M	Diploma in Nursing and Midwifery	2009
h12	KCH	Tertiary Hospital	27	F	Six year Medical degree (MBBS)	2009
h13	Mangochi	District Hospital – Secondary	27	M	Six year Medical degree (MBBS)	2010

Table I.5: Health worker interview responses - part B

Participant	Used Computers in college	What for?	Use CDS tools during consultation	Evidence of CPG text	Use digital devices for your work	What for?	How long?	Current challenges?
h1	No	N/A	Yes	No	No	N/A	N/A	N/A
h2	No	N/A	Yes	No	No	N/A	N/A	N/A
h3	Yes	Research, Report writing	Yes	Yes	No	N/A	N/A	N/A
h4	Yes	Research, Assignments	Yes	Yes	No	N/A	N/A	N/A
h5	Yes	Email, research, data collection	Yes	Yes	No	N/A	N/A	N/A
h6	Yes	For research but with very limited resources	Yes	Yes	Yes, but not regularly	Capturing complaints, diagnosis and treatment	2 years	Slows consultation process
h7	Yes	Typing assignments, listening to music, computer games	Yes	Yes	No	N/A	N/A	N/A
h8	Yes, in a way	Only had one computer lesson	Yes	Yes	No	N/A	N/A	N/A
h9	No	N/A	Yes	Yes	Yes, partially	Prescribing and diagnosis	2 years	A limited system without full diagnosis and prescription functionality
h10	No	N/A	Yes	Yes	Yes, but not regularly	Captures diagnosis and treatment	1 year	Slows work down
h11	No	N/A	Yes	Yes	No	N/A	N/A	N/A
h12	Yes	Reading, searching for info online, analysing data	Yes	Yes	No	N/A	N/A	N/A
h13	Yes	Research, studying, assignments	Yes	Yes	Yes	References and research e.g Medskip	4 Years	Most guidelines that are electronically available are different to local guidelines

Table I.6: Health worker interview responses - part C

Participant	**How often do you refer to CPG text?**	**Use digital devices for social activities?**	**Which ones?**	**Possible to use tech during consultation?**	**How?**
h1	Not very often	Yes	Mobile FB	Yes	To store medical history. But can slow people down, frequent breakdown of machines for current system
h2	Not very often	Yes	Mobile FB	Yes	To store medical history. But Need enough devices and standard treatment plans in the systems
h3	Often (2-3 days) per week	Yes	FB, E-mail	Yes	To store medical history, to lookup reference material and guidelines
h4	Often (2-3 days) per week	Yes	FB, Twitter, E-mail	Yes	To store medical history
h5	Often (2-3 days) per week	Yes	FB, Twitter, E-mail	Yes	To store medical history, research, retrieve patient info
h6	Very often (More than 3 times a week)	Yes	Facebook, online news feeds	Yes	To refer to guidelines for difficult conditions. Current system captures very limited information, not very useful
h7	Very often (More than 3 times a week)	Yes	Music, games, digital pictures	Yes	To store data, to search for diagnostic procedures e.g. internet references; probable management of cases
h8	Often (2-3 days) per week	Yes	Facebook, E-mail, MS Powerpoint, Access	Yes	Keeping medical history
h9	Very often (More than 3 times a week)	Yes	Saving, copying and printing text documents. Listening and sharing music and pictures. Social networking on Facebook	Not sure	It may take years as it wouldn't help you much. Ideal for data collection but not for patient care.
h10	Often (2-3 days) per week	Yes	Computer games	Yes	To store records, remind them of guidelines
h11	Very often (More than 3 times a week)	Yes	FB, e-mail, Whatsup	Yes	Best practices pulled from digital devices
h12	Often (2-3 days) per week	Yes	Social networks, e-mail, video calls	Yes	To store medical records, To refer to guidelines, Video conferencing
h13	Very often (More than 3 times a week)	Yes	Facebook, twitter, football news	Yes	Used for easy reference of material as as to prevent preventable mistakes

Appendix J

CPG change incidents

J.1 Malawi HIV CPG changes in 2011

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
Providing ART in special situations	ART in case of renal failure	N	Y	Change in inclusion criteria	additional decision variable
Prescribing CPT	Any child, aged 6 weeks or more, born to an HIV-positive woman	N	Y	Change in inclusion criteria	additional decision variable
Managing suspected ART drug failure	Presenting with suspected ART drug failure	N	Y	Change in inclusion criteria	additional decision variable
∼	∼	∼	∼	∼	3
Definition of ART eligibility	Adults [persons aged 15 years and above] - WHO Clinical Stage 3 or WHO Clinical Stage 4	N	Y	Change in inclusion criteria	Change in decision criteria value

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Dis-con-tin-ued**	**Cha-nged**	**Change Type**	**Specific change**
	Adults [persons aged 15 years and above] - CD4 lymphocyte below 250/mm3	N	Y	Change in inclusion criteria	Change in decision criteria value
	Children over the age of 18 months - CD4 or TLC counts (<3 yrs)	N	Y	Change in inclusion criteria	Change in decision criteria value
	Children over the age of 18 months - CD4 or TLC counts (3 years to <5 yrs)	N	Y	Change in inclusion criteria	Change in decision criteria value
	PSHD - OC and SP	N	Y	Change in inclusion criteria	Change in decision criteria value
	PSHD – OC and SS	N	Y	Change in inclusion criteria	Change in decision criteria value
	PSHD – SP and SS	N	Y	Change in inclusion criteria	Change in decision criteria value
	Children over the age of 18 months - WHO Paediatric Clinical Stage 4 or WHO Paediatric Clinical Stage 3	N	Y	Change in inclusion criteria	Change in decision criteria value
Managing first-line regimen in children	Triomune-Baby	N	Y	Change in inclusion criteria	Change in decision criteria value
Determining standard adult doses of ART drugs	Zidovudine	N	Y	Change in inclusion criteria	Change in decision criteria value

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Stavudine	N	Y	Change in inclusion criteria	Change in decision criteria value
	Abacavir	N	Y	Change in inclusion criteria	Change in decision criteria value
	Tenofovir	N	Y	Change in inclusion criteria	Change in decision criteria value
∼	∼	∼	∼	∼	13
Providing ART in special situations	D4T/3TC/NVP can be given to lactating mothers	N	Y	Change in inclusion criteria	drop of decision variable
Managing suspected ART drug failure	Suspected ART failure – new WHO stage 4	N	Y	Change in inclusion criteria	drop of decision variable
∼	∼	∼	∼	∼	2
Prescribing CPT	Dosages of CPT – Children aged 6 weeks to 5 months	N	Y	Change in inclusion criteria	new decision variable
	Dosages of CPT – Children aged 6 months to 4 years	N	Y	Change in inclusion criteria	new decision variable
	Dosages of CPT – Children aged 5- 14 years	N	Y	Change in inclusion criteria	new decision variable
	Dosages of CPT – Adults	N	Y	Change in inclusion criteria	new decision variable
	child has stopped breast- feeding	N	Y	Change in inclusion criteria	new decision variable
Standard Adult Doses of Antiretroviral Drugs	Zidovudine	N	Y	Change in inclusion criteria	new decision variable

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	Recommendation	Discontinued	Changed	Change Type	Specific change
	Stavudine	N	Y	Change in inclusion criteria	new decision variable
	Abacavir	N	Y	Change in inclusion criteria	new decision variable
	Tenofovir	N	Y	Change in inclusion criteria	new decision variable
	Efavirenz	N	Y	Change in inclusion criteria	new decision variable
	Lopinavir / ritonavir	N	Y	Change in inclusion criteria	new decision variable
Managing suspected ART drug failure	Suspected ART failure – new WHO stage 4	N	Y	Change in inclusion criteria	new decision variable
∼	∼	∼	∼	∼	12
∼	∼	∼	∼	30	0
Providing ART in special situations	ART in case of renal failure	N	Y	Change in recommended action	additional recommended action
Prescribing CPT	pregnant woman is on CPT	N	Y	Change in recommended action	additional recommended action
∼	∼	∼	∼	∼	2
Determining standard adult doses of ART drugs	Zidovudine	N	Y	Change in recommended action	change in action verb complement
	Stavudine	N	Y	Change in recommended action	change in action verb complement
	Abacavir	N	Y	Change in recommended action	change in action verb complement

TABLE J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Tenofovir	N	Y	Change in recommended action	change in action verb complement
∼	∼	∼	∼	∼	4
Providing ART in special situations	D4T/3TC/NVP can be given to lactating mothers	N	Y	Change in recommended action	change in action verb complement
∼	∼	∼	∼	∼	1
Providing ART in special situations	D4T/3TC/NVP can be given to lactating mothers	N	Y	Change in recommended action	drop of recommended action
Providing ART in special situations	ART in case of renal failure	N	Y	Change in recommended action	drop of recommended action
∼	∼	∼	∼	∼	2
Providing ART in special situations	Patients with acute hepatitis	N	Y	Change in recommended action	new recommended action
Managing suspected ART drug failure	Presenting with suspected ART drug failure	N	Y	Change in recommended action	new recommended action
∼	∼	∼	∼	∼	2
∼	∼	∼	∼	11	0
Prescribing CPT	Any child, 6 weeks or more, who is HIV-positive	N	N	∼	∼
Definition of ART eligibility	Infant <12 months - positive DNA-PCR	N	N	∼	∼
Managing ART patients who develop TB	If there is any suspicion of TB	N	N	∼	∼
∼	∼	∼	∼	∼	0

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
∼	∼	∼	∼	0	0
∼	∼	44	∼	∼	∼
Standard Adult Doses of Antiretroviral Drugs	Lamivudine	Y	N/A	∼	∼
	Nevirapine	Y	N/A	∼	∼
Implementing standardised ART reviews	Side effects between reviews	Y	N/A	∼	∼
	After 6 months	Y	N/A	∼	∼
	After one year	Y	N/A	∼	∼
Managing first-line drug reactions	Severe peripheral neuropathy	Y	N/A	∼	∼
	Pancreatitis	Y	N/A	∼	∼
	lactic acidosis/ Lipodystrophy syndrome	Y	N/A	∼	∼
	Skin reactions	Y	N/A	∼	∼
	Hepatitis	Y	N/A	∼	∼
	Adverse reactions to first line regimen - child <3 yrs	Y	N/A	∼	∼
Providing ART in special situations	Contraceptives and Neverapine	Y	N/A	∼	∼
	Contraceptives and Efavirenz	Y	N/A	∼	∼
	d4T/3TC/NVP is not contraindicated in pregnancy	Y	N/A	∼	∼
	At the onset of labour - discontinue NVP	Y	N/A	∼	∼
	Child born to HIV positive mother on ART	Y	N/A	∼	∼

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Patients with established stable chronic liver disease	Y	N/A	~	~
	ART eligible patients with tuberculosis	Y	N/A	~	~
	initial phase of anti-TB treatment	Y	N/A	~	~
	Initial phase of anti-TB treatment - severely immunocompromised patients	Y	N/A	~	~
	continuation phase of anti-TB treatment with rifampicin and isoniazid	Y	N/A	~	~
Prescribing CPT	DNA-PCR- negative children	Y	N/A	~	~
	All patients eligible for ART	Y	N/A	~	~
	CD4 count monitoring - Adults	Y	N/A	~	~
	Discontinuing CPT - adults - severe cutaneous reactions	Y	N/A	~	~
	Discontinuing CPT - adults - renal or hepatic toxicity	Y	N/A	~	~
	Discontinuing CPT - adults - severe haematological	Y	N/A	~	~

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Dis-con-tin-ued**	**Cha-nged**	**Change Type**	**Specific change**
	Discontinuing CPT - severe cutaneous reactions	Y	N/A	~	~
	Discontinuing CPT - renal or hepatic toxicity	Y	N/A	~	~
	Discontinuing CPT - severe haematological toxicity	Y	N/A	~	~
	Person with symptomatic HIV disease	Y	N/A	~	~
	Any person – CD4 count of 500/mm 3 or less	Y	N/A	~	~
	Pregnant women irrespective of duration of pregnancy	Y	N/A	~	~
Definition of ART eligibility	Presumed severe HIV disease (PSHD) - advanced HIV disease in the mother	Y	N/A	~	~
	Presumed severe HIV disease (PSHD) - CD4 <20% in children 12-18 months	Y	N/A	~	~

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Presumed severe HIV disease (PSHD) - CD4 <25% in children less than 12 months	Y	N/A	~	~
	Presumed severe HIV disease (PSHD) - recent HIV-related maternal death	Y	N/A	~	~
	At 18 months - If the test is positive - On ART	Y	N/A	~	~
	At 18 months - If the test is negative - On ART	Y	N/A	~	~
	Children under the age of 18 months - WHO Paediatric Clinical Stage 3 or 4 (>= 12 months)	Y	N/A	~	~
	Children under the age of 18 months - CD4 or TLC counts below threshold values for starting ART (>= 12 months)	Y	N/A	~	~
	Children over the age of 18 months - CD4 or TLC counts (5 years to <15 years)	Y	N/A	~	~

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Adults [persons aged 15 years and above] - WHO Clinical Stage 2 with a total lymphocyte count <1200/mm3	Y	N/A	~	~
	Children under the age of 18 months - WHO Paediatric Clinical Stage 4	Y	N/A	~	~
Managing first-line regimen in children	T30 - Triomune baby unavailable	Y	N/A	~	~
	T30	Y	N/A	~	~
Managing Kaposi 's sarcoma	In patients with mild to moderate disease	Y	N/A	~	~
	For others (severe)	Y	N/A	~	~
Managing ART patients who develop TB	If TB develops when the patient is on ART	Y	N/A	~	~
	If patients on second line ART (which contains protease inhibitors) develop TB	Y	N/A	~	~
	If patients on second line ART develop drug-resistant TB	Y	N/A	~	~
Monitoring and Managing Drug Toxicity	Management of Pancreatitis – diagnosis	Y	N/A	~	~

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Management of peripheral neuropathy - diagnosis	Y	N/A	~	~
	Management of peripheral neuropathy - treatment	Y	N/A	~	~
	Management of peripheral neuropathy - treatment (No response after 4 weeks)	Y	N/A	~	~
Determining priority for CD4 count testing	HIV-positive pregnant women	Y	N/A	~	~
	Patients in WHO stage 2	Y	N/A	~	~
	HIV-infected children	Y	N/A	~	~
	Whenever ART failure is suspected	Y	N/A	~	~
	Base-line	Y	N/A	~	~
	Follow-up every 12 months	Y	N/A	~	~
	Follow-up every 6 months	Y	N/A	~	~
	Follow-up every 6-12 months	Y	N/A	~	~
Standard Adult Doses of Antiretroviral Drugs	Didanosine for patients >= 60kg	Y	N/A	~	~
	Didanosine for patients <60kg	Y	N/A	~	~

Table J.1: Malawi HIV CPG change incidents in 2011 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Zalcitabine	Y	N/A	∼	∼
	Emtricitabine	Y	N/A	∼	∼
	Nelfinavir	Y	N/A	∼	∼
	Saquinavir / ritonavir	Y	N/A	∼	∼
	Indinavir / ritonavir	Y	N/A	∼	∼
	Atazanavir	Y	N/A	∼	∼
	Atazanavir/ritonavir	Y	N/A	∼	∼
Managing suspected ART drug failure	Suspected ART Drug Failure - CD4 count / CD4% - pre-treatment values or less	Y	N/A	∼	∼
	new or worsening symptoms since last visit e.g.(fever, abnominal pain, vomiting, diarrhoea, weight loss, rash, pain)	Y	N/A	∼	∼

J.2 Malawi HIV CPG changes in 2014

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
Choosing regimen and time of starting in special situations	Anaemia	N	Y	Change in inclusion criteria	additional decision variable
	Active TB >= 3yrs and <15 yrs	N	Y	Change in inclusion criteria	additional decision variable
∼	∼	∼	∼	∼	2
Choosing regimen and time of starting in special situations	Active TB >= 3yrs and <15 yrs	N	Y	Change in inclusion criteria	change in decision criteria value
Definition of ART eligibility	Child 12 to under 24 months – HIV rapid antibody test or DNA–PCR	N	Y	Change in inclusion criteria	change in decision criteria value
	Child or adult 5 years and over – Pregnant women	N	Y	Change in inclusion criteria	change in decision criteria value
	Child or adult 5 years and over – WHO stage 1 or 2	N	Y	Change in inclusion criteria	change in decision criteria value
∼	∼	∼	∼	∼	4
Choosing regimen and time of starting in special situations	Renal failure (6 wks to <15yrs)	N	Y	Change in inclusion criteria	drop of decision variable
	Renal failure (>15 yrs)	N	Y	Change in inclusion criteria	drop of decision variable
∼	∼	∼	∼	∼	2
CD4 monitoring for ART eligibility	CD4 monitoring of patients in HIV Care Clinic follow–up – with confirmed HIV infection	N	Y	Change in inclusion criteria	new decision variable

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Dis-con-tin-ued**	**Cha-nged**	**Change Type**	**Specific change**
Choosing regimen and time of starting in special situations	Active TB – Under 3 yrs	N	Y	Change in inclusion criteria	new decision variable
	Active TB >= 15 yrs	N	Y	Change in inclusion criteria	new decision variable
	Jaundice - initiation (6wks to 14 yrs)	N	Y	Change in inclusion criteria	new decision variable
	Jaundice - initiation (15 yrs+)	N	Y	Change in inclusion criteria	new decision variable
~	~	~	~	~	5
~	~	~	~	13	~
Choosing regimen and time of starting in special situations	Anaemia	N	Y	Change in recommended action	change in action verb complement
	Jaundice - initiation (15 yrs+)	N	Y	Change in recommended action	change in action verb complement
	Renal failure (6 wks to <15yrs)	N	Y	Change in recommended action	change in action verb complement
	Renal failure (>15 yrs)	N	Y	Change in recommended action	change in action verb complement
~	~	~	~	~	4
~	~	~	~	4	~
ART side-effects - stopping ART	Yellow eyes / hepatitis	N	N	~	~
	Severe stomach pain and vomiting	N	N	~	~
	Shortness of breath	N	N	~	~

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Severe skin rash with blisters, involving eyes, mouth or genitals	N	N	∼	∼
CD4 monitoring for ART eligibility	CD4 monitoring of patients in HIV Care Clinic follow–up – Stopping CD4 monitoring	N	N	∼	∼
Choosing regimen and time of starting in special situations	Jaundice – referral	N	N	∼	∼
	In labour (new HIV+)	N	N	∼	∼
Clinical suspicion and diagnosis of treatment failure	suspected ART failure diagnosis	N	N	∼	∼
	suspected ART failure with good adherence	N	N	∼	∼
	suspected ART failure with poor adherence	N	N	∼	∼
	VL 5,000 copies/ml or more	N	N	∼	∼
Combining ART and TB treatment	Do not combine without specialist advice	N	N	∼	∼
Cotrimoxazole preventive therapy (CPT)	All infants born to HIV infected mothers	N	N	∼	∼
	Confirmed HIV infected children from age 6 weeks and adults	N	N	∼	∼

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Giving SP to HIV infected pregnant women on CPT	N	N	∼	∼
	If SP has already been taken - pregnant women	N	N	∼	∼
	Who and when to stop on CPT - HIV exposed children	N	N	∼	∼
	Who and when to stop on CPT - severe side effects	N	N	∼	∼
	Dosage of Cotrimoxazole Preventive Therapy - less than 6kg	N	N	∼	∼
	Dosage of Cotrimoxazole Preventive Therapy - 6.0 – 13.9kg	N	N	∼	∼
	Dosage of Cotrimoxazole Preventive Therapy - 14.0 – 29.9kg	N	N	∼	∼
	Dosage of Cotrimoxazole Preventive Therapy - 30.0kg and above	N	N	∼	∼
Definition of ART eligibility	Infant under 12 months – Confirmed HIV infection (DNA–PCR needed)	N	N	∼	∼

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Presumed severe HIV disease (PSHD) - oral candidiasis and severe pneumonia	N	N	~	~
	Presumed severe HIV disease (PSHD) - oral candidiasis and severe sepsis	N	N	~	~
	Presumed severe HIV disease (PSHD) - severe pneumonia and severe sepsis	N	N	~	~
	Presumed severe HIV disease (PSHD) - Pneumocystis pneumonia	N	N	~	~
	Presumed severe HIV disease (PSHD) - Candidiasis of oesophagus, trachea, bronchi or lungs	N	N	~	~
	Presumed severe HIV disease (PSHD) - Cryptococcal meningitis	N	N	~	~

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
	Presumed severe HIV disease (PSHD) - Severe unexplained wasting / malnutrition not responding to treatment (weight-for-height/ -age <70% or MUAC <11cm or oedema)	N	N	∼	∼
	Presumed severe HIV disease (PSHD) - Toxoplasmosis of the brain (from age 1 month)	N	N	∼	∼
	Child or adult 5 years and over - breastfeeding women	N	N	∼	∼
	Child or adult 5 years and over - WHO clinical stage 3 or 4	N	N	∼	∼
Indications for interrupting or stopping ART	Lactic acidosis	N	N	∼	∼
	Pancreatitis	N	N	∼	∼
	chronic poor adherence	N	N	∼	∼
Isoniazid preventive therapy (IPT)	Eligibility for IPT	N	N	∼	∼
	Stopping IPT	N	N	∼	∼
Monitoring of nutritional status	Weight–for–height less than 80% – Children 0–14 years	N	N	∼	∼

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	Recommendation	Discontinued	Changed	Change Type	Specific change
	MUAC less than 12cm – Children 0–14 years	N	N	∼	∼
	Start ART if no response to TF after 3 weeks	N	N	∼	∼
	Weight loss >10% – Non–pregnant adults 15 years and above	N	N	∼	∼
	BMI under 18.5 – Non–pregnant adults 15 years and above	N	N	∼	∼
	BMI under 17 – Non–pregnant adults 15 years and above	N	N	∼	∼
	BMI under 16 – Non–pregnant adults 15 years and above	N	N	∼	∼
	Universally eligible for ART – Pregnant or lactating women	N	N	∼	∼
	MUAC less than 22cm	N	N	∼	∼
	MUAC less than 19cm	N	N	∼	∼
	Use MUAC instead of BMI	N	N	∼	∼

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
Provider initiated family planning (PIFP)	Implementing routine PIFP in HIV clinic - men	N	N	∼	∼
	Implementing routine PIFP in HIV clinic - women	N	N	∼	∼
	Giving Depo–Provera	N	N	∼	∼
Provider initiated testing and counselling (PITC)	never tested	N	N	∼	∼
	tested negative more than 3 months ago	N	N	∼	∼
	claims to have been tested any time in the past, but without documentation	N	N	∼	∼
Routine ascertainment of HIV exposure status for children under 24 months	For the child: If the mother is not available / has died	N	N	∼	∼
	For the child: If the mother is not available / has died	N	N	∼	∼
	For the child: If the child is sick, even if the mother was tested negative during pregnancy or delivery	N	N	∼	∼
Routine TB screening	standard screening questions	N	N	∼	∼

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Dis-con-tin-ued**	**Cha-nged**	**Change Type**	**Specific change**
Selecting regimen and formulation for continuation	All children who were on the new standard 1st line paediatric regimen (AZT / 3TC / NVP, Regimen 2)	N	N	~	~
	Children on 1st line regimens when their weight is over 25kg	N	N	~	~
	Children who were on paediatric 2nd line regimen (Regimen 9P)	N	N	~	~
~	~	~	~	~	0
~	~	~	~	0	~
~	~	79	~	~	~
Definition of ART eligibility	Child 24 months to under 5 years - Confirmed HIV infection (HIV rapid antibody test) and WHO stage 1 or 2	Y	N/A	~	~
	Child 24 months to under 5 years - WHO clinical stage 3 or 4	Y	N/A	~	~
Selecting regimen and formulation for continuation	If a woman became pregnant while on an ART regimen that contains EFV	Y	N/A	~	~
~	~	~	~	~	0
~	~	~	~	0	~

Table J.2: Malawi HIV CPG change incidents in 2014 .

Guideline	**Recommendation**	**Discontinued**	**Changed**	**Change Type**	**Specific change**
∼	∼	3	∼	∼	∼
∼	∼	82	∼	∼	∼
∼	∼	∼	∼	∼	∼

Appendix K

Grammar usability questionnaire responses

K.1 SUS questionnaire responses

Table K.1: SUS questionnaire responses - Novice CIG modellers .

Participant	Use grammar frequently	Grammar was complex	Grammar easy to use	Need support of a technical person	Functions well integrated	Too much inconsistency	Learn to use grammar quickly	Grammar very cumbersome	Felt confident	Needed to learn a lot
p1	3	3	4	4	4	2	3	2	3	2
p2	3	2	3	2	4	2	4	3	3	2
p3	4	2	3	4	4	2	2	1	4	2
p4	4	1	3	2	3	1	5	1	3	2
p5	4	1	5	5	4	1	5	2	4	1
p6	3	1	5	1	4	1	4	1	1	1
p7	4	1	4	3	5	1	2	1	5	2
p8	3	1	5	1	5	1	2	1	2	1
p9	4	2	5	2	4	1	4	2	5	2
p10	5	2	4	1	4	2	5	1	4	2
p11	5	1	5	1	5	1	5	1	3	3

Table K.1: SUS questionnaire responses - Novice CIG modellers .

Participant	**Use grammar frequently**	**Grammar was complex**	**Grammar easy to use**	**Need support of a technical person**	**Functions well integrated**	**Too much inconsistency**	**Learn to use grammar quickly**	**Grammar very cumbersome**	**Felt confident**	**Needed to learn a lot**
p12	5	1	5	1	4	1	3	1	5	1
p13	4	1	5	1	5	1	4	1	5	2

Table K.2: SUS questionnaire responses - Experienced CIG modellers .

Participant	**Use grammar frequently**	**Grammar was complex**	**Grammar easy to use**	**Need support of a technical person**	**Functions well integrated**	**Too much inconsistency**	**Learn to use grammar quickly**	**Grammar very cumbersome**	**Felt confident**	**Needed to learn a lot**
p1	5	1	3	2	3	1	3	1	4	3
p2	3	1	5	2	4	1	5	1	5	1
p3	3	1	4	1	4	1	5	1	5	1
p4	4	1	5	1	4	1	5	1	4	1
p5	4	1	4	1	5	1	5	1	4	1
p6	5	1	4	1	4	1	5	1	5	1

K.2 Qualitative questionnaire responses

Table K.3: Qualitative questionnaire responses - Novice CIG Modellers .

Partici-pant	**Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?**	**Are there any features or keywords that are missing or need to be improved in the grammar?**
p1		No
p2	action, condition, recommendation	
p3	conditons, actions	N/A
p4	Action	Not that I know of
p5	the three constructs -¿ explain the model well. specifically Condition as it can load data that's vital to making an informed recommendation	A concern is that Recommendation may have many entries for conditions for very specific cases making the syntax long an cumbersome , the possibility of creating a grouping of conditions may help in such complex cases. Just from a readability perspective
p6	Action	RECOMMENDER-to explain where a recommendation comes from(who/which body/doctor/nurse/organisation). Actions should be able to be grouped (into sets of actions). Actions should. Also does the order of actions matter? can I assume actions are sequencial? CONDITION – HIV_STAGE IS ADVANCED and ACTION PRESCRIPE-NO HIV_MEDICATION, They both have HIV noun, NOUN should be extractable in action and condition for searchability, linking and possibly other uses of clinical practice guideline.
p7	confidence/confident	
p8	The constructs	Not the I'm aware.

Table K.3: Qualitative questionnaire responses - Novice CIG Modellers .

Partici-pant	**Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?**	**Are there any features or keywords that are missing or need to be improved in the grammar?**
p9	all of them are necessary	Have the usefulness of concepts such as pre-conditions, assertions been investigated?
p10	It was quite idiomatic and the syntax flows.	What if a reccomendation is based on another reccommendation? Possibly have the possiblity for
p11		
p12	Conditions, Action and Recommendations	Choosing between singular and plural can be tricky.e.g. in case of one condition, I thought conditions will change to Condition
p13	Found the condition construct and its breakdown into units and decision variable very useful and highly informative	The UML class diagram initially give me the impression of two part activity diagram. It was only after going through that I understood it as one part. Can you modify the diagram and have the Guideline at the top and rest flowing below so as to avoid this.

Table K.4: Qualitative questionnaire responses - Experienced CIG Modellers .

Partici-pant	**Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?**	**Are there any features or keywords that are missing or need to be improved in the grammar?**
Participant	Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?	Are there any features or keywords that are missing or need to be improved in the grammar?

Table K.4: Qualitative questionnaire responses - Experienced CIG Modellers .

Partici-pant	**Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?**	**Are there any features or keywords that are missing or need to be improved in the grammar?**
p1	condition, action, recommendation. These really cover the basics one would need to use in this kind of setup	Not sure if there could be an explanation later of how the language can be linked to an interpreting engine and how they will be expected to work together. I feel an implementer can easily appreciate the value of the language if they also have a view on how the language can be used in a practical environment but overall this is a very relevant tool.
p2	clinical practice guideline, computer-interpretable guidelines, action, condition, recommendation	where writing a recommendation, are the action and conditions always compulsory or are they sometimes optional?
p3	It was all pretty intuitive. I appreciated the definitions of the key models and how few of them there were and how well they fit together.	I think the ability to define more complex logic might be necessary in many situations. e.g. this_variable = this_other_thing + 30 or a = b and c or (d and e). I understand this is (somewhat) handled by the concept dictionary now, but that information seems like it might need to be in the grammar/protocol itself
p4	Relator, operators, condition/action/recommendation names	1. Definition of qualified/authorised actors/users. 2. Specifying how the action should be performed (recommended tools, methods) where necessary 3. Specifying when (timeframe) the action should be performed where necessary. These can be specified for a group of guidelines
p5	The syntax , CAR for Conditions , Actions and Recommendations were easy to follow	
p6	condition	None

Appendix L

FCIG against Arden Syntax raw data

L.1 FCIG against Arden Syntax measurements

TABLE L.1: FCIG against Arden Syntax measurements - part A .

Participant	Language	Time on task 1 (seconds)					Time on task 2 (in seconds)						
		Sub task 1	Sub task 2	Sub task 3	Total	Mean	Sub task 1	Sub task 2	Sub task 3	Sub task 4	Sub task 5	Sub task 6	Sub task 7
p2	Arden Syntax	867	467	346	1680	560	203	158	113	220	154	105	160
p3	Arden Syntax	489	235	264	988	329.3333333333	123	116	109	190	95	75	79
p4	Arden Syntax	684	189	209	1082	360.6666666667	183	42	27	71	86	27	34
p5	Arden Syntax	916	555	229	1700	566.6666666667	288	167	116	270	118	88	58
p6	Arden Syntax	856	265	192	1313	437.6666666667	237	174	95	184	103	63	118
p8	Arden Syntax	899	151	205	1255	418.3333333333	94	97	64	118	96	65	90
p9	Arden Syntax	704	234	273	1211	403.6666666667	174	258	97	159	104	104	87

Table L.1: FCIG against Arden Syntax measurements - part A .

Participant	**Language**	**Time on task 1 (seconds)**					**Time on task 2 (in seconds)**						
		Sub task 1	**Sub task 2**	**Sub task 3**	**Total**	**Mean**	**Sub task 1**	**Sub task 2**	**Sub task 3**	**Sub task 4**	**Sub task 5**	**Sub task 6**	**Sub task 7**
p10	Arden Syntax	947	155	163	1265	421.6666666667	159	138	145	66	89	66	83
p11	Arden Syntax	829	474	302	1605	535	212	168	84	289	89	130	64
p12	Arden Syntax	900	135	240	1275	425	90	90	60	100	130	90	100
p13	Arden Syntax	1026	407	244	1677	559	161	122	107	239	102	124	124
p14	Arden Syntax	1070	363	193	1626	542	204	255	60	266	188	158	107
p15	Arden Syntax	1212	232	196	1640	546.6666666667	177	164	73	73	124	172	19
p16	Arden Syntax	1021	237	179	1437	479	131	87	86	261	141	90	85
p17	Arden Syntax	973	327	258	1558	519.3333333333	230	227	70	194	110	107	103
p18	Arden Syntax	1700	480	285	2465	821.6666666667	215	218	220	170	123	135	148
p19	Arden Syntax	715	225	220	1160	386.6666666667	155	132	49	95	194	119	81
p20	Arden Syntax	917	320	224	1461	487	167	193	262	217	170	157	106
p21	Arden Syntax	1137	384	248	1769	589.6666666667	211	239	108	429	246	297	180
p23	Arden Syntax	923	216	214	1353	451	345	55	40	53	141	181	60
p24	Arden Syntax	908	527	333	1768	589.3333333333	296	361	170	125	109	104	117
p25	Arden Syntax	1590	470	324	2384	794.6666666667	200	130	102	255	175	128	208

Table L.1: FCIG against Arden Syntax measurements - part A .

Participant	Language	**Time on task 1 (seconds)**					**Time on task 2 (in seconds)**						
		Sub task 1	**Sub task 2**	**Sub task 3**	**Total**	**Mean**	**Sub task 1**	**Sub task 2**	**Sub task 3**	**Sub task 4**	**Sub task 5**	**Sub task 6**	**Sub task 7**
p26	Arden Syntax	533	330	273	1136	378.6666666667	283	212	54	85	48	127	86
p27	Arden Syntax	900	397	208	1505	501.6666666667	191	138	315	86	146	126	114
p29	Arden Syntax	729	269	254	1252	417.3333333333	155	97	138	125	975	157	119
p31	Arden Syntax	900	168	161	1229	409.6666666667	114	100	32	102	55	71	76
p2	FCIG	438	373	232	1043	347.6666666667	256	265	97	87	56	24	89
p3	FCIG	568	281	214	1063	354.3333333333	257	186	64	126	40	31	60
p4	FCIG	467	257	176	900	300	139	119	14	50	14	31	21
p5	FCIG	570	532	497	1599	533	281	259	142	169	103	114	72
p6	FCIG	379	295	115	789	263	162	153	28	88	26	34	45
p8	FCIG	612	299	170	1081	360.3333333333	177	142	53	39	59	37	46
p9	FCIG	356	283	133	772	257.3333333333	205	157	49	66	57	56	49
p10	FCIG	823	285	187	1295	431.6666666667	252	182	26	72	70	26	41
p11	FCIG	440	360	258	1058	352.6666666667	283	369	61	79	50	65	60
p12	FCIG	437	190	120	747	249	80	85	22	60	25	25	32
p13	FCIG	479	336	277	1092	364	305	267	47	60	35	75	54
p14	FCIG	1112	630	282	2024	674.6666666667	298	185	181	45	67	75	29
p15	FCIG	711	269	141	1121	373.6666666667	176	174	25	43	66	22	39
p16	FCIG	248	288	198	734	244.6666666667	132	151	17	61	56	4	30
p17	FCIG	299	299	214	812	270.6666666667	202	131	40	68	66	75	40
p18	FCIG	782	430	235	1447	482.3333333333	284	142	38	103	108	70	62
p19	FCIG	532	152	163	847	282.3333333333	183	102	32	29	37	32	50
p20	FCIG	616	330	228	1174	391.3333333333	172	218	70	115	99	125	95
p21	FCIG	649	443	204	1296	432	297	258	58	169	168	135	91
p23	FCIG	613	399	336	1348	449.3333333333	137	307	71	183	103	150	72
p24	FCIG	559	528	313	1400	466.6666666667	206	137	117	74	71	49	63

Table L.1: FCIG against Arden Syntax measurements - part A .

Participant	**Language**	**Time on task 1 (seconds)**					**Time on task 2 (in seconds)**						
		Sub task 1	**Sub task 2**	**Sub task 3**	**Total**	**Mean**	**Sub task 1**	**Sub task 2**	**Sub task 3**	**Sub task 4**	**Sub task 5**	**Sub task 6**	**Sub task 7**
p25	FCIG	985	390	193	1568	522.6666666667	290	368	55	140	150	63	60
p26	FCIG	1034	551	278	1863	621	91	114	75	38	242	64	76
p27	FCIG	589	360	223	1172	390.6666666667	255	261	98	68	48	22	254
p29	FCIG	410	265	156	831	277	134	137	37	55	21	25	41
p31	FCIG	736	287	266	1289	429.6666666667	211	163	46	25	46	91	90

Table L.2: FCIG against Arden Syntax measurements - part B .

Participant	**Language**	**Time on task 2 (in seconds)**					**SUS Scores**	**Task 1**		**Task 2**		**Efficiency on task**	
		Sub task 8	**Sub task 9**	**Sub task 10**	**Total**	**Mean**		**ELOC**	**LOC**	**ELOC**	**LOC**	**Task 1**	**Task 2**
p2	Arden Syntax	551	89	42	1795	179.5	67.5	136	176	447	561	0.000595238095	0.000557103064
p3	Arden Syntax	33	38	20	878	87.8	35	138	177	449	559	0.001012145749	0.001138952164
p4	Arden Syntax	21	31	9	531	53.1	60	136	173	453	556	0.000924214418	0.001883239171
p5	Arden Syntax	104	171	38	3398	339.8	95	136	177	432	549	0.000588235294	0.000294290759
p6	Arden Syntax	122	209	22	1327	132.7	92.5	130	170	431	541	0.000761614623	0.000753579503
p8	Arden Syntax	47	61	13	745	74.5	62.5	130	154	433	543	0.000796812749	0.001342281879
p9	Arden Syntax	41	104	36	1164	116.4	82.5	130	165	432	542	0.000825763832	0.000859106529
p10	Arden Syntax	49	100	35	930	93	50	127	159	446	557	0.000790513834	0.001075268817

TABLE L.2: FCIG against Arden Syntax measurements - part B .

		Time on task 2 (in seconds)						**Task 1**		**Task 2**		**Efficiency on task**	
Participant	**Language**	**Sub task 8**	**Sub task 9**	**Sub task 10**	**Total**	**Mean**	**SUS Scores**	**ELOC**	**LOC**	**ELOC**	**LOC**	**Task 1**	**Task 2**
p11	Arden Syntax	44	141	11	1232	123.2	92.5	139	186	433	545	0.00062305296	0.000811688312
p12	Arden Syntax	20	68	18	766	76.6	82.5	127	158	432	543	0.000784313725	0.001305483029
p13	Arden Syntax	57	120	47	1203	120.3	85	139	170	435	547	0.000596302922	0.000831255195
p14	Arden Syntax	21	108	32	1399	139.9	87.5	137	178	449	560	0.00061500615	0.000714796283
p15	Arden Syntax	15	20	73	910	91	80	130	162	450	562	0.000609756098	0.001098901099
p16	Arden Syntax	41	177	28	1127	112.7	80	130	164	435	548	0.000695894224	0.000887311446
p17	Arden Syntax	100	115	32	1288	128.8	87.5	133	170	432	542	0.000641848524	0.000776397516
p18	Arden Syntax	98	67	50	1444	144.4	30	140	185	489	616	0.000405679513	0.000692520776
p19	Arden Syntax	24	77	16	942	94.2	62.5	136	169	432	542	0.000862068966	0.001061571125
p20	Arden Syntax	65	220	11	1568	156.8	70	139	174	484	605	0.000684462697	0.000637755102
p21	Arden Syntax	43	116	28	1897	189.7	67.5	136	179	436	548	0.000565291125	0.000527148129
p23	Arden Syntax	11	38	19	943	94.3	75	132	164	443	553	0.0007390983	0.001060445387
p24	Arden Syntax	44	207	39	1572	157.2	90	151	198	451	563	0.00056561086	0.000636132316
p25	Arden Syntax	52	66	14	1330	133	80	135	169	451	563	0.000419463087	0.000751879699
p26	Arden Syntax	30	149	31	1105	110.5	87.5	140	178	434	547	0.00088028169	0.000904977376

TABLE L.2: FCIG against Arden Syntax measurements - part B .

Participant	**Language**	**Time on task 2 (in seconds)**					**SUS Scores**	**Task 1**		**Task 2**		**Efficiency on task**	
		Sub task 8	**Sub task 9**	**Sub task 10**	**Total**	**Mean**		**ELOC**	**LOC**	**ELOC**	**LOC**	**Task 1**	**Task 2**
p27	Arden Syntax	29	65	20	1230	123	70	132	183	447	557	0.000664451827	0.00081300813
p29	Arden Syntax	18	72	13	1869	186.9	72.5	133	175	432	542	0.000798722045	0.000535045479
p31	Arden Syntax	44	51	27	672	67.2	47.5	129	160	448	563	0.00081366965	0.001488095238
p2	FCIG	21	100	22	1017	101.7	82.5	34	64	85	164	0.000958772771	0.000983284169
p3	FCIG	26	128	11	929	92.9	55	31	46	84	160	0.000940733772	0.001076426265
p4	FCIG	13	89	12	502	50.2	82.5	31	39	84	153	0.001111111111	0.001992031873
p5	FCIG	22	203	21	1386	138.6	32.5	33	66	79	147	0.000625390869	0.000721500722
p6	FCIG	34	82	7	659	65.9	82.5	34	56	84	155	0.001267427123	0.001517450683
p8	FCIG	23	109	27	712	71.2	85	31	45	84	157	0.00092506938	0.001404494382
p9	FCIG	24	85	8	756	75.6	40	31	52	84	154	0.001295336788	0.001322751323
p10	FCIG	18	137	12	836	83.6	92.5	31	51	84	155	0.000772200772	0.001196172249
p11	FCIG	37	177	12	1193	119.3	97.5	31	58	84	158	0.000945179584	0.000838222967
p12	FCIG	15	33	9	386	38.6	92.5	31	46	82	154	0.001338688086	0.002590673575
p13	FCIG	12	179	6	1460	146	90	34	62	84	157	0.000915750916	0.000684931507
p14	FCIG	5	182	71	1198	119.8	87.5	33	55	84	158	0.000494071146	0.000834724541
p15	FCIG	27	82	31	685	68.5	92.5	31	60	84	164	0.00089206066	0.001459854015
p16	FCIG	44	185	6	686	68.6	80	34	53	84	156	0.00136239782	0.001457725948
p17	FCIG	18	118	15	773	77.3	87.5	31	50	84	155	0.001231527094	0.001293661061
p18	FCIG	30	170	11	1018	101.8	65	31	51	84	155	0.000691085003	0.000982318271
p19	FCIG	28	131	5	629	62.9	82.5	29	43	84	158	0.001180637544	0.001589825119
p20	FCIG	55	183	31	1163	116.3	87.5	34	58	79	147	0.000851788756	0.000859845228
p21	FCIG	68	172	4	4780	478	62.5	31	60	84	159	0.000771604938	0.000209205021
p23	FCIG	25	57	15	1120	112	72.5	31	34	82	148	0.000741839763	0.000892857143
p24	FCIG	29	153	18	917	91.7	87.5	39	72	88	163	0.000714285714	0.001090512541
p25	FCIG	56	110	30	1322	132.2	65	29	42	82	156	0.000637755102	0.000756429652
p26	FCIG	40	100	44	884	88.4	90	32	63	82	152	0.000536768653	0.001131221719

Table L.2: FCIG against Arden Syntax measurements - part B .

Participant	Language	Time on task 2 (in seconds)					SUS Scores	Task 1		Task 2		Efficiency on task	
		Sub task 8	Sub task 9	Sub task 10	Total	Mean		ELOC	LOC	ELOC	LOC	Task 1	Task 2
p27	FCIG	16	122	21	1165	116.5	85	31	50	84	155	0.000853242321	0.000858369099
p29	FCIG	14	74	5	543	54.3	90	31	47	84	158	0.001203369434	0.001841620626
p31	FCIG	38	156	31	1497	149.7	75	31	55	84	157	0.00077579519	0.000668002672

Table L.3: Error rates .

Participant	Language	Task 1				Task 2										
		t1	t2	t3	Overall	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	Overall
p2	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p3	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p4	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p5	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p6	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p8	Arden Syntax	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
p9	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p10	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p11	Arden Syntax	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
p12	Arden Syntax	1	1	1	1	0	0	0	0	0	0	0	0	1	0	1
p13	Arden Syntax	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Table L.3: Error rates .

		Task 1				**Task 2**										
Participant	**Language**	**t1**	**t2**	**t3**	**Overall**	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**t8**	**t9**	**t10**	**Overall**
p14	Arden Syntax	1	1	0	1	0	0	0	1	1	0	0	0	1	0	1
p15	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p16	Arden Syntax	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1
p17	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p18	Arden Syntax	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
p19	Arden Syntax	0	1	0	1	0	1	0	1	1	1	0	0	0	0	1
p20	Arden Syntax	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p21	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p23	Arden Syntax	0	0	0	0	1	1	0	0	0	1	0	0	1	0	1
p24	Arden Syntax	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1
p25	Arden Syntax	1	1	1	1	1	0	1	0	0	0	1	0	1	0	1
p26	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
p27	Arden Syntax	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
p29	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p31	Arden Syntax	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
p2	FCIG	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1
p3	FCIG	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
p4	FCIG	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Table L.3: Error rates .

Participant	Language	Task 1				Task 2										
		t1	**t2**	**t3**	**Overall**	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**t8**	**t9**	**t10**	**Overall**
p5	FCIG	0	0	1	1	0	1	0	1	0	1	1	0	0	0	1
p6	FCIG	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
p8	FCIG	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
p9	FCIG	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1
p10	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
p11	FCIG	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1
p12	FCIG	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1
p13	FCIG	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1
p14	FCIG	1	1	1	1	0	1	1	0	1	1	1	1	1	0	1
p15	FCIG	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
p16	FCIG	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1
p17	FCIG	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
p18	FCIG	1	1	1	1	0	0	0	1	0	0	0	0	0	1	1
p19	FCIG	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
p20	FCIG	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
p21	FCIG	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
p23	FCIG	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1
p24	FCIG	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1
p25	FCIG	1	1	0	1	0	0	0	1	1	0	0	0	1	0	1
p26	FCIG	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1
p27	FCIG	0	1	1	1	0	0	1	1	1	1	1	0	0	0	1
p29	FCIG	1	1	1	1	0	1	1	0	0	0	1	0	0	0	1
p31	FCIG	1	1	1	1	0	1	1	0	1	1	0	0	0	0	1

Table L.4: Number of errors .

Participant	Language	Task 1					Tast 2											
		t1	**t3**	**t3**	**total**	**avg**	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**t8**	**t9**	**t10**	**total**	**avg**
p2	Arden Syntax	0	1	1	2	0.67	1	0	0	0	1	0	1	1	1	0	5	0.5

Table L.4: Number of errors .

Participant	**Language**	**Task 1**					**Tast 2**											
		t1	**t3**	**t3**	**total**	**avg**	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**t8**	**t9**	**t10**	**total**	**avg**
p3	Arden Syntax	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0.1
p4	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0.1
p5	Arden Syntax	0	0	1	1	0.33	0	1	0	1	0	1	2	0	0	0	5	0.5
p6	Arden Syntax	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.1
p8	Arden Syntax	0	1	0	1	0.33	0	0	0	0	0	0	0	0	0	0	0	0
p9	Arden Syntax	0	1	1	2	0.67	0	0	0	0	0	1	0	0	0	0	1	0.1
p10	Arden Syntax	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.1
p11	Arden Syntax	0	0	1	1	0.33	0	0	0	0	0	1	0	0	0	0	1	0.1
p12	Arden Syntax	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	2	0.2
p13	Arden Syntax	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	3	0.3
p14	Arden Syntax	1	1	2	4	1.33	0	2	2	0	1	1	1	1	1	0	9	0.9
p15	Arden Syntax	1	0	0	1	0.33	0	0	0	0	0	0	0	0	0	0	0	0
p16	Arden Syntax	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	2	0.2
p17	Arden Syntax	0	1	0	1	0.33	0	0	0	0	0	0	0	0	0	0	0	0
p18	Arden Syntax	1	1	1	3	1	0	0	0	1	0	0	0	0	0	1	2	0.2
p19	Arden Syntax	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0.1

TABLE L.4: Number of errors .

Participant	Language	Task 1					Tast 2											
		t1	t3	t3	total	avg	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	total	avg
p20	Arden Syntax	0	1	1	2	0.67	1	1	1	1	1	1	1	1	1	1	10	1
p21	Arden Syntax	0	1	0	1	0.33	0	0	0	0	0	0	0	0	0	0	0	0
p23	Arden Syntax	2	2	2	6	2	0	1	1	1	1	1	1	0	1	0	7	0.7
p24	Arden Syntax	1	0	0	1	0.33	0	0	0	0	0	0	0	0	1	0	1	0.1
p25	Arden Syntax	1	1	0	2	0.67	0	0	0	1	1	0	0	0	1	0	3	0.3
p26	Arden Syntax	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	5	0.5
p27	Arden Syntax	0	1	1	2	0.67	0	0	1	1	1	1	1	0	0	0	5	0.5
p29	Arden Syntax	1	1	1	3	1	0	1	1	0	0	0	1	0	0	0	3	0.3
p31	Arden Syntax	1	1	1	3	1	0	1	1	0	1	1	0	0	0	0	4	0.4
p2	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p3	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p4	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p5	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p6	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p8	FCIG	1	1	1	3	1	0	0	0	0	0	0	0	0	0	0	0	0
p9	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p10	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p11	FCIG	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0.1
p12	FCIG	1	1	1	3	1	0	0	0	0	0	0	0	0	1	0	1	0.1
p13	FCIG	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0.1
p14	FCIG	1	1	0	2	0.67	0	0	0	1	1	0	0	0	1	0	3	0.3
p15	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p16	FCIG	1	0	0	1	0.33	0	0	0	0	1	0	0	0	0	0	1	0.1

Table L.4: Number of errors .

Participant	Language	Task 1					Tast 2											
		t1	**t3**	**t3**	**total**	**avg**	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**t8**	**t9**	**t10**	**total**	**avg**
p17	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p18	FCIG	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0.1
p19	FCIG	0	1	0	1	0.33	0	1	0	1	1	1	0	0	0	0	4	0.4
p20	FCIG	0	0	1	1	0.33	0	0	0	0	0	0	0	0	0	0	0	0
p21	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p23	FCIG	0	0	0	0	0	1	1	0	0	0	1	0	0	2	0	5	0.5
p24	FCIG	1	0	0	1	0.33	0	1	0	0	0	0	0	0	0	0	1	0.1
p25	FCIG	1	1	1	3	1	1	0	1	0	0	0	1	0	1	0	4	0.4
p26	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.1
p27	FCIG	0	1	1	2	0.67	0	0	0	0	0	0	0	0	0	0	0	0
p29	FCIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p31	FCIG	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0.1

L.2 SUS questionnaire responses

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	Use language frequently	Comments	Language complex	Comments	Language easy to use	Comments	Need support of a technical person	Comments
p2	4		2		4		2	

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p3	2	Not maintainable. It would require fixed meta-labels on the recommendations to recall what they are. This is currently done by the recommendation's name but it is insufficient. A tool for graphical editing (with autocomplete) would improve my view.	3		4	It might be complex for beginners, especially ones with no programming experience.	4	A few training sessions for users would be necessary.
p4	4	Easy to understand	2	When the conditions are all clamped up it gets a little bit disorienting	5		2	
p5	1	requires too much care and attention to detail to the user	5	The language structure seems ok, but is not that easy to use but it could get better with practice	2	It takes some getting used to	3	To get familiar with the bits and pieces

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p6	4		2		5		1	
p8	5	Flexible, easy to use and modify	1	Language is very intuitive	5	Could use linking words and comparators (¡, ¿) which make it very easy to understand	4	Some training would be necessary, but I think anyone would be able to use this language
p9	3	It would not be my first choice if I had to choose.	4	It is a bit hard to understand at first.	2	It took me sometime to get started.	4	Only in the beginning.
p10	5	Takes some time to wrap mind around the syntax but once one gets going its a peace of cake	1		5		2	

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p11	4		1	It is very simple and easy to use	5	I thought it was extremely easy to use, with the exception of not knowing exactly what the prescribed variables are. I knew age was a prescribe variable, but I didn't know if height was, or what qualified a variable to be a variable to be a variable.	1	
p12	5		2		5		1	
p13	5		1		4		2	
p14	4		2		5		2	

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p15	4	straight forward, pleasant enough.	1	The language quite natural. nothing confusing about it for me. I was overly concerned with form.	5		1	I am very familiar with the technical aspects of such languages
p16	4	yes, editing this language i think was a bit easier since you reused conditions a lot	2	there was a lot of work to be done in the beginning, with setting up conditions and actions, but after that it was quite easy	4		2	
p17	5		1		5		1	
p18	2		2		3		4	

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p19	4	It was very quick to understand, a person with no programming experience can adapt to this language easily, I wish there was a library to avoid spending time on typing words.	1	The only thing complex was keeping track to the numbering of recommendation/condition/action as if i had to edit a recommendation i would take a huge time trying to find the particular recommendation	5	Very quick to understand as it does not require any programming experience	2	I would need someone only in the first few minutes to understand the syntax of the language otherwise
p20	5	very reusable	1		4		2	
p21	4		2		2		3	
p22	5	It's very is to understand and use	1	It's very easy to understand	5		1	
p23	5		3	The editing of the conditions is a bit tricky	2		1	Only a person who understand the language is needed for first time use explanation.

Table L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p24	5	feels natural and informal	1	only what is required is written	5	has high code reusability	1	easy to pick up
p25	3	It was a learning curve at first, but after a while it became easier to use and made more sense.	2	It seemed so at first, but it's actually fairly simple.	3		2	I may need help but not all the time.
p26	5	The language is straight forward using simple sentences to describe a function or method	1		3	I had a bit of trouble at first getting the concept but as i went on i managed to get a better idea of how to go on	3	Just a bit of help in getting a better idea on what to do next and when
p27	2		2		4		1	

TABLE L.5: SUS questionnaire responses - *FCIG* (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p29	5	The language is extremely intuitive and has the potential to significantly improve clinical recommendation systems.	2	The language is fairly easy to pick up and get used to at first glance.	4	The language syntax is very intuitive and fairly simple.	1	I think that if you don't have experience in programming, you could need some help at first but as you continue using the programming language, it becomes intuitive and simple to use.
p31	4		2		4		2	

TABLE L.6: SUS questionnaire responses - *FCIG* (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p2	5		1		4		2	
p3	4		1		3	Health workers in rural areas might have a difficult time using it. It then depends on the training offered.	4	
p4	4	the recommendation, action and condition structure is simple	1		4	very high level language	1	
p5	3	Everything was too clustered	3	the language was fairly consistend	2	A Big No No	4	Very cumbersome even though it might not seem that way
p6	4		2		4		1	

TABLE L.6: SUS questionnaire responses - *FCIG* (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p8	5	Autocomplete functions really helped with finding conditions etc. This language was well integrated with Eclipse	2	The wide range of naming freedom such as	5	Those with basic computer literacy would not struggle with this language. It is very intuitive.	1	The separation of conditions, actions and recommendations makes it a very succinct language that is easy to modify
p9	2	When the code is large enough, It will be hard to keep track of your actions, recommendations and conditions.	2	You can use the same variable many times.	3		4	Everything is all in one place and you can not group things together.
p10	5		1		4		1	
p11	5		1		5		1	
p12	4		1		5		1	
p13	4		1		4		1	
p14	4		2		5		1	

Table L.6: SUS questionnaire responses - *FCIG* (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p15	4	I kind of want a way to quickly go back to the Recommendation after Ctrl+Click ing on the conditions/actions	1		5		2	Descriptive variable names are cumbersome in any language
p16	4		1		4		1	

Table L.6: SUS questionnaire responses - *FCIG* (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p17	4		2		5	There are very few keywords to learn and the document follows an intuitive, simple structure	4	Conditions cannot have multiple clauses. If we could use "and' and 'or' operators to compose conditions, it might declutter the list of conditions a little. having to workaround this limitation by using 2 conditions is a little cumbersome
p18	4		1	Well-structured, though it is technical	4	People with some knowledge of programming can pick it up quickly	2	Structure is simple enough to understand

Table L.6: SUS questionnaire responses - *FCIG* (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p19	3	Having being told that who the language is designed for, I would have expected functions and libraries for the particular target.	2		5		2	
p20	5		1		3		1	
p21	3		2		4	Although the language has limitations in capturing recommendations, it is easy to use and learn.	2	
p22	5		2		5		1	
p23	5		2		3		2	
p24	4	reading patients info is well integrated	5	found if very consistent	5		1	it's function orientated. not much structural code is needed

Table L.6: SUS questionnaire responses - *FCIG* (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p25	5		4	Because things could be defined differently, it could be confusing when using the different names.	4		3	
p26	5		1		5	The language is way easier then language A once you get a good idea of it and how it works	1	
p27	4		1		5		1	
p29	5		1	The language is very consistent in terms of its structure and it will be fairly simple to read other people's program structure.	5		1	
p31	5		2		4		2	

Table L.7: SUS questionnaire responses - *FCIG* (part C)

Participant	Felt confident	Comments	Needed to learn a lot	Comments	Any useful feautures?	Any missing features?
p2	5		2		code re-use	
p3	4		3			
p4	4	Not sure what the drug names were until the end	2		The python like structure, use of semicolons with no constrains on the positions of the leading white spaces	composite conditions (having conditions within conditions) may make it compact
p5	3	My confidence dropped a bit	3	The syntax	All	Not really
p6	4		2		Reusability	None
p8	4	After some practise, it became very easy to use	2	With the help of the wiki and the spreadsheet, I didn't need to know anything about the medical conditions in order to get to grips with it.	Autocomplete was very helpful, the outline allowed me to see what conditions had already been created.	Not that I can think of
p9	2	I was confused.	2	There isn't a lot to learn to use this language.	Using variables you define once and use multiple times	Code grouping. Group Conditions together, Group Recommendations together and Group Actions together.
p10	5		2		The outline	

p11	5		1		The actions are very simple to use and the fact that it only has conditions, actions and recommendations makes it even easier to use.	No.
p12	5		2		Reuse of conditions in different recommendations	the ability to select ranges (eg 1¡x¡5)
p13	5		1			
p14	5		1		scripting from the same file/page	all the basic mathematical operands should be catered for as widely know. eg. the between x and y as one expression as opposed to separate entity
p15	5		1		all of them. Although I did not use the Ctrl+hover feature, and chose to Ctrl+click (see above). That would have been wiser. Silly me for not paying attention	I don't know :—
p16	5		3		The reusing of conditions and actions was very helpful	no

p17	5		1	Extremely simple, straightforward syntax was easy to learn. Very few keywords that were intuitive and easy-to-learn document structure	Reusing the conditions and actions was a really good feature that saved a lot of time. It also made the recommendations very easy to parse (I could completely understand what was going on even when reading the example cig file for the first time).	Being able to compose conditions and/or use multiple clauses to define a condition would be nice.
p18	4		2		References to certain conditions, statements	Being able to put the list of conditions and list of recommendations in the different sections
p19	5	Very easy to understand	2	Only to get started otherwise, after the experiment i feel as if i can use this language on my own	the shortcuts as it prevented me to type the redundant text	It needed a sort of library for clinical guidelines to avoid typing, for example, instead of typing positive or negative it could already be installed in the language.
p20	4		1		The fact that we could reuse the conditions	Not that I can think of
p21	4		3			

p22	5		2		Separating the conditions and the actions. It made the modelling easy to understand.	Nothing I can think of
p23	4		2		Having outlined variable links on the IDE makes it easy to navigate to the actual definitions of the variables hence making it easy and fast to use.	
p24	5	tag names helps with navigation	1		it's function orientated. not much structural code is needed	
p25	3	At first no, but after a while I became more confident	1		The auto-completion and suggestions when filling in conditions and actions	Not to my knowledge
p26	5	Easy to understand and implement	1		All the features are useful like in language A	
p27	5		1		Ease of use in adding various conditions or actions to guide a patients recommendation	

p29	4		2	I needed to learn a few medical jargon and the basic syntax of the programming language, however, it becomes intuitive even with minimum medical knowledge.	Referencing Condition and Action constructs to make recommendations is a very useful feature, and significantly minimizes human error in my opinion.	
p31	4		3		Autocomplete very useful, Overview also helpful	Consistency with condition variables could be better. I know for example

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p2	3		2		4		5	
p3	2	Maintenance is easier. However, the language is too verbose. It is also hard to keep track of individual mlm's since they are all in a single file.	4		2		4	

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p4	2	The detailed information is appreciated however the use of threshold values for each recommendation is redundant.	4		3		2	
p5	5	The language is easy to use	1	Once you become familiar with the syntax everything starts to fall in place	5	The language is straight forward	1	
p6	5		1		5		1	
p8	4	More complex than Language A, but allows for reading from a record which I think would be very useful.	2	Some fields like	4	Again it was fairly intuitive but separating the data (variable declaration) and logic components made it slightly less intuitive	5	I definitely needed guidance to begin using this language.

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p9	5	The language is straight forward. It is easy to follow whats happening.	1	The language is easy to understand and I did not find it complex.	5	It has the basic concepts of other programming languages and that makes it easy.	5	This experiment got me interesting in working with this programming language.
p10	3		3		3		4	
p11	5	It's very easy to use and learn as well.	1		5		1	
p12	4		2	Not complex but repetitive, would be nice to define some variables once and use in multiple places.	5		1	
p13	4		2		4		1	
p14	4		1		4		2	
p15	3	I'd prefer the other one, but I got used to the MLM towards the end	2	It seems complex in relation to the CIG. But it isn't really.	4	Simple enough once I got the hang of it. And I got the hang of it very fast	1	Once again, it was easy after I got the hang of it

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p16	4	It was very easy to learn after reading the wiki	1		5	the wiki and spreadsheet really helped in making things easy	3	
p17	5		1		5		1	
p18	2		4		1		4	Very technical, would need some guidance in explaining the syntax used

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	Use language frequently	Comments	Language complex	Comments	Language easy to use	Comments	Need support of a technical person	Comments
p19	3	Its a good language for a programmer but someone who has never done programming will definitely find it difficult to understand	2	It was very easy and fast to edit the document although if i wasn't given a template at first, i would've struggled	3	If i wasn't given a template of how to do one patient I would've had a lot of errors and difficulty	4	It would be unfair for a person who has never programmed to use this language as they would struggle to fix serious errors if they mess up the template or they don't know the value of a semicolon.
p20	4		1		4		3	
p21	5		4		4		2	
p23	5		1		3		3	Technical support will be needed for people not familiar with programming

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p24	5	easy to pickup	2	the use of semi colons can be tricky at first	5	tags on the side help with navigation, syntax is close to english	1	might need some help with variable names for data access
p25	5	I'd use this more often because it is nicely structured and more readable. It also gives more information under the different sections.	2		4		2	
p26	4	The language is straight forward and easy to implement with little back ground knowledge.	1		4	It took only one example to get the basic idea of how the language should be done.	1	
p27	3		1		5		3	

Table L.8: SUS questionnaire responses - Arden Syntax (part A)

Participant	**Use language frequently**	**Comments**	**Language complex**	**Comments**	**Language easy to use**	**Comments**	**Need support of a technical person**	**Comments**
p29	4		2	The language is fairly intuitive.	4	The language uses an easy-to-understand syntax.	2	A technical person could be required when first trying out the language. It will be fairly easy to use without additional assistance on a day-to-day basis.
p31	3		4		3		2	

Table L.9: SUS questionnaire responses - Arden Syntax (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p2	3		2		3		1	
p3	3		1		1		5	

Table L.9: SUS questionnaire responses - Arden Syntax (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p4	5	The if construct is well integrated in the logic section.	1		2	takes some time to getting used to.	3	
p5	4	They get the job done	1		5	The language is not complex	1	
p6	4		1		4		1	
p8	5	Once again the autocomplete function was very useful. This language would have been far more challenging and time-consuming to use without it.	1	I felt that Language B is more consistent than Language A, at the expense of its flexibility.	1	I think this system would take some time to get used to, especially when executing complex functions.	3	A little cumbersome
p9	3	It feels easy to use already.	1		5		1	
p10	4		1		2		3	
p11	5	I found they were integrated in a way that contributed to the efficiency of the language.	1		5		1	
p12	5		2		5		2	
p13	5		1		5		2	
p14	4		1		5		1	

Table L.9: SUS questionnaire responses - Arden Syntax (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p15	5		1		2	The language is more technical than most, and closely resembles other programming languages. These always take users some time to learn, particularly if they are unfamiliar with languages such as these	2	At first yes, but very quickly it became light and easy to use
p16	4		1		4		1	

Table L.9: SUS questionnaire responses - Arden Syntax (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p17	5		4	Case insensitivity can be a little confusing	5		2	I'm not sure why logic and action were separated. Why not just add the action where we currently write 'conclude TRUE' -¿ this is boilerplate code that doesn't really encode any valuable information.
p18	4		2		2		4	Very technical, with a lot of detail

Table L.9: SUS questionnaire responses - Arden Syntax (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p19	4	It was a lot quicker to edit although I kept losing track of where to edit.	1	It was well consistent, there was very little to edit	3	This language could be good for people who are interested in learning programming but it would definitely take time to understand especially if i wasn't given a template.	1	It was very fast to use especially in editing
p20	3	Did not really notice functions and integration	1		2	If the person does not have a programming background, it may not be easy	2	
p21	5		4		4		2	
p23	4		2		3		2	
p24	3		1	the same, simple structure is used throughout	5	the structure is intuitive	2	
p25	4		2		3		1	

Table L.9: SUS questionnaire responses - Arden Syntax (part B)

Participant	**Functions well integrated**	**Comments**	**Too much inconsistency**	**Comments**	**Learn language very quickly**	**Comments**	**Very cumbersome**	**Comments**
p26	4	The functions were similar to other programming languages which made it simple to implement.	1		5	As long as you get told how the language works and go through an example of the language than the language is easy to learn quickly	1	
p27	4		2		1		2	
p29	4		1	The language was consistent and has a specific programming structure that makes it easy to get used to it.	3	It would take a reasonable amount of time to get used to the programming language if you are a beginner at programming, however it can be very intuitive in due time even with the most inexperienced programmer.	2	It is easy to debug if errors have occurred because the structure of the language is intuitive.
p31	3		2		2		4	

Table L.10: SUS questionnaire responses - Arden Syntax (part C)

Participant	**Felt confident**	**Comments**	**Needed to learn a lot**	**Comments**	**Any useful feautures?**	**Any missing features?**
p2	5		1			object oriented approach to some of th data types
p3	4		4			
p4	5	The detail provided in the language increased my confidence in the language	3		The if statement in the logic section.	linking recommendations together.
p5	4	As a first time user I have to say that it was not that bad, it was actually easy to use.	1	No	All	I think the language has what it is needed for the task at hand.
p6	5		2		The language uses commonly used operators for logic	None
p8	4	After some time I became comfortable, especially with the editing section. Descriptions such as purpose and explanation sections make it time consuming to edit	2	It took longer to get to grips with than language A, but was fairly intuitive once I became familiar with the layout.	Autocomplete, referring to the wiki for examples	Possibly an autocomplete for reference record database names to make it easier to refer to the records.

p9	5		2		Having separate blocks. e.g maintenance, Library and Knowledge	Not that I can think of.
p10	2		3		The outline	
p11	5		4	You just have to know the operators and have working knowledge of variables and variable declaration.	I found the description very useful, and it made navigating the code easy together with the outline.	No.
p12	4		3		Autocomplete	None come to mind
p13	4		2			
p14	5		2			
p15	5	At first not, but then yes	1		Ctrl+space	Not sure
p16	5		4	Even though was a lot to learn, the documentation helped make the process easy	the auto-complete made filling in the blocks (maintenance etc) very easy.	None, that i can think of
p17	4	May have been more confident with better domain knowledge	1		Having multiple mlm's in the same file was useful -¿ made it easier to keep related modules together. Allowing comments helped clarify complex bits.	

p18	2		5		library section for referencing patients	Explanation of how to refer to medical records in the data section
p19	4	If i was given plenty of time to practice i would eventually be good at it	4		I like how you can move from record to record using the outline toolbar.	I wish there was an interface that would display the output of the code. And i wish there I was a
p20	4		2		the readability and having many mlm in one file	Not that I can think of
p21	4		3		use of variables and a more flexible way of specifying logic rules.	Re-use of data across slots.
p23	4		1		Defining constants before using them and reading data from spreadsheet	
p24	5	auto complete + underlining errors helps you code quickly	1	all it takes is a 10 min read through and a reference manual handy	The tag structure and easy to read syntax	
p25	5		2	There was not much to learn, very simple and easy operations	The descriptions and variable assigning.	

p26	5	It's straight forward and easy to pick up	3	I might need a bit more information on what i can and can not do in the language just to see what i can do with it	I find keywords very useful in specifying values or data to use once i head to the logic part of the program to run the actual instruction i want.	
p27	4		1			
p29	4		3	There is a reasonable amount of knowledge regarding the program language structure that is required in order to get going with the language, but it is intuitive.		

p31	4		4		Again the standard IDE Jumping and autocomplete were useful.	However the rewriting of conditions was frustrating and the overall feel of the language felt too verbose. The Jumping via the overview also took you to the end of the relevant module after which you needed to scroll up which was a little confusing. Syntax was also slightly over complicated.

Bibliography

[1] Ulrich Frank. Domain-specific modeling languages: requirements analysis and design guidelines. In *Domain Engineering*, pages 133–157. Springer, 2013.

[2] Yamiko J. Msosa, C. Maria Keet, and Melissa Densmore. Characterisation of clinical practice guideline changes. In *9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016)*, volume 5 of *HEALTHINF*, pages 248–255, 2016.

[3] Malawi Ministry of Health. Health sector strategic plan. Technical report, The Malawi Government, 2011.

[4] Rony Zachariah, Nathan Ford, Mitt Philips, S. Lynch, M. Massaquoi, V. Janssens, and A. D. Harries. Task shifting in HIV/AIDS: opportunities, challenges and proposed actions for sub-Saharan Africa. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 103(6):549–558, 2009. ISSN 0035-9203. doi: 10.1016/j.trstmh.2008.09.019.

[5] UNDP. Human Development Report 2016: Human development for everyone. Technical report, The United Nations Development Programme, 2016.

[6] Daniel A. Handel and Jeffrey L. Hackman. Implementing electronic health records in the emergency department. *Journal of Emergency Medicine*, 38(2):257–263, 2010.

[7] Faustine Williams and Suzanne Austin Boren. The role of the electronic medical record (EMR) in care delivery development in developing countries: a systematic review. *Informatics in Primary Care*, 16(2):139–145, 2008.

[8] Mit Philips, Rony Zachariah, and Sarah Venis. Task shifting for antiretroviral treatment delivery in sub-Saharan Africa: not a panacea. *The Lancet*, 371(9613): 682–684, 2008. ISSN 01406736. doi: 10.1016/S0140-6736(08)60307-4.

[9] Zach Landis-Lewis, Claudia Mello-Thoms, Oliver J. Gadabu, Miranda E. Gillespie, Gerald P. Douglas, and Rebecca S. Crowley. The feasibility of automating audit

and feedback for ART guideline adherence in Malawi. *Journal of the American Medical Informatics Association*, 18(6):868–874, 2011. ISSN , 1527-974X. doi: 10.1136/amiajnl-2011-000097.

[10] Karen M. Lyng. From clinical practice guidelines, to clinical guidance in practice - impacts for computerization. *International Journal of Medical Informatics*, 82 (12):358–363, 2013.

[11] Mor Peleg. Computer-interpretable clinical guidelines: A methodological review. *Journal of Biomedical Informatics*, 46(4):744–763, 2013. ISSN 1532-0464. doi: 10.1016/j.jbi.2013.06.009.

[12] Paul de Clercq, Katharina Kaiser, and Arie Hasman. Computer-interpretable guideline formalisms. *Studies in Health Technology and Informatics*, 139:22–43, 2008.

[13] Matt-Mouley Bouamrane, Alan Rector, and Martin Hurrell. Using OWL ontologies for adaptive patient information modelling and preoperative clinical decision support. *Knowledge and information systems*, 29(2):405–418, 2011.

[14] Katrina F. Hurley and Syed S. R. Abidi. Ontology engineering to model clinical pathways: Towards the computerization and execution of clinical pathways. In *Computer-Based Medical Systems, 2007. CBMS'07. Twentieth IEEE International Symposium on*, pages 536–541. IEEE, 2007.

[15] Veruska Zamborlini, Marcos da Silveira, Cédric Pruski, Annette ten Teije, and Frank van Harmelen. Towards a conceptual model for enhancing reasoning about clinical guidelines. In Silvia Miksch, David Riaño, and Annette ten Teije, editors, *Knowledge Representation for Health Care*, number 8903 in Lecture Notes in Computer Science, pages 29–44. Springer International Publishing, 2014. ISBN 978-3-319-13280-8 978-3-319-13281-5.

[16] Yamiko J. Msosa, Melissa Densmore, and C. Maria Keet. Towards an architectural design of a guideline-driven EMR system: A contextual inquiry of Malawi. In *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development*, ICTD '15, pages 49:1–49:4. ACM, 2015. ISBN 978-1-4503-3163-0. doi: 10.1145/2737856.2737899.

[17] Raquel Walker. Walking Beyond Our Borders With Frontline Health Workers in Guatemala. *Nursing for Women's Health*, 17(6):533–538, December 2013. ISSN 1751-486X. doi: 10.1111/1751-486X.12082.

[18] Smisha Agarwal, Henry B. Perry, Lesley-Anne Long, and Alain B. Labrique. Evidence on feasibility and effective use of mHealth strategies by frontline health workers in developing countries: systematic review. *Tropical Medicine & International Health*, 20(8):1003–1014, August 2015. ISSN 1365-3156. doi: 10.1111/tmi.12525.

[19] Burke R. Johnson and Anthony J. Onwuegbuzie. Mixed methods research: A research paradigm whose time has come. *Educational researcher*, 33(7):14–26, 2004.

[20] John W. Creswell and Vicki L. Plano Clark. Designing and conducting mixed methods research. 2007.

[21] John W. Creswell. *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research.* Pearson, 2011. ISBN 978-0-13-136739-5.

[22] David W. Bates, Gilad J. Kuperman, Samuel Wang, Tejal Gandhi, Anne Kittler, Lynn Volk, Cynthia Spurr, Ramin Khorasani, Milenko Tanasijevic, and Blackford Middleton. Ten commandments for effective clinical decision support: Making the practice of evidence-based medicine a reality. *Journal of the American Medical Informatics Association*, 10(6):523–530, 2003. doi: 10.1197/jamia.M1370.

[23] Amir Qaseem, Frode Forland, Fergus Macbeth, Günter Ollenschläger, Sue Phillips, and Philip van der Wees. Guidelines International Network: toward international standards for clinical practice guidelines. *Annals of internal medicine*, 156(7): 525–531, 2012.

[24] Richard M. Rosenfeld, Richard N. Shiffman, and Peter Robertson. Clinical practice guideline development manual: a quality-driven approach for translating evidence into action. *Otolaryngology—Head and Neck Surgery*, 148(1_suppl):S1–S55, 2013.

[25] Thomas Knoll, Muhammad I. Omar, Steven Maclennan, Virginia Hernández, Steven Canfield, Yuhong Yuan, Max Bruins, Lorenzo Marconi, Hein Van Poppel, James N'Dow, and Richard Sylvester. Key Steps in Conducting Systematic Reviews for Underpinning Clinical Practice Guidelines: Methodology of the European Association of Urology. *European Urology*, September 2017. ISSN 0302-2838. doi: 10.1016/j.eururo.2017.08.016.

[26] William Rosenberg and Anna Donald. Evidence based medicine: an approach to clinical problem-solving. *BMJ: British Medical Journal*, 310(6987):1122, 1995.

[27] David L. Sackett, William M. C. Rosenberg, J. A. Muir Gray, R. Brian Haynes, and W. Scott Richardson. Evidence based medicine: what it is and what it isn't. *BMJ*, 312(7023):71–72, January 1996. ISSN 0959-8138, 1468-5833. doi: 10.1136/bmj.312.7023.71.

[28] David L. Sackett. Evidence-based medicine. *Seminars in Perinatology*, 21(1):3–5, 1997. ISSN 0146-0005. doi: 10.1016/S0146-0005(97)80013-4.

[29] Brian R. Haynes. What kind of evidence is it that Evidence-Based Medicine advocates want health care providers and consumers to pay attention to? *BMC Health Services Research*, 2:3, 2002. ISSN 1472-6963. doi: 10.1186/1472-6963-2-3.

[30] Trisha Greenhalgh. *How to read a paper: The basics of evidence-based medicine.* John Wiley & Sons, 2014.

[31] Paul Garner, Martin Meremikwu, Jimmy Volmink, Qian Xu, and Helen Smith. Putting Evidence Into Practice: How Middle And Low Income Countries "Get It Together". *BMJ: British Medical Journal*, 329(7473):1036–1039, 2004. ISSN 0959-8138.

[32] Paul Chinnock, Nandi Siegfried, and Mike Clarke. Is Evidence-Based Medicine Relevant to the Developing World? *PLOS Medicine*, 2(5):e107, May 2005. ISSN 1549-1676. doi: 10.1371/journal.pmed.0020107.

[33] Zulma Ortiz, Pablo Perel, and Jordi Pardo. Evidence-Based Medicine in Iberoamerica: Problems and Possible Solutions. *PLOS Medicine*, 2(8):e277, August 2005. ISSN 1549-1676. doi: 10.1371/journal.pmed.0020277.

[34] Celia McMichael, Elizabeth Waters, and Jimmy Volmink. Evidence-based public health: what does it offer developing countries? *Journal of Public Health*, 27(2): 215–221, June 2005. ISSN 1741-3842. doi: 10.1093/pubmed/fdi024.

[35] Dominique Behague, Charlotte Tawiah, Mikey Rosato, Télésphore Some, and Joanna Morrison. Evidence-based policy-making: The implications of globally-applicable research for context-specific problem-solving in developing countries. *Social Science & Medicine*, 69(10):1539–1546, November 2009. ISSN 0277-9536. doi: 10.1016/j.socscimed.2009.08.006.

[36] Cristina Tomatis, Claudia Taramona, Emiliana Rizo-Patrón, Fiorela Hernández, Patricia Rodríguez, Alejandro Piscoya, Elsa Gonzales, Eduardo Gotuzzo, Gustavo Heudebert, Robert M. Centor, and Carlos A. Estrada. Evidence-based medicine training in a resource-poor country, the importance of leveraging personal and institutional relationships. *Journal of Evaluation in Clinical Practice*, 17(4):644–650, August 2011. ISSN 1365-2753. doi: 10.1111/j.1365-2753.2011.01635.x.

[37] Taryn Young, Paul Garner, Mike Clarke, and Jimmy Volmink. Series: Clinical Epidemiology in South Africa. Paper 1: Evidence-based health care and policy in Africa: past, present, and future. *Journal of Clinical Epidemiology*, 83(Supplement C):24–30, March 2017. ISSN 0895-4356. doi: 10.1016/j.jclinepi.2016.06.006.

[38] Richard Grol. Successes and Failures in the Implementation of Evidence-Based Guidelines for Clinical Practice. *Medical Care*, 39(8):II46–II54, August 2001.

[39] Richard Grol and Jeremy Grimshaw. From best evidence to best practice: effective implementation of change in patients' care. *The Lancet*, 362(9391):1225–1230, October 2003. ISSN 0140-6736. doi: 10.1016/S0140-6736(03)14546-1.

[40] Marcia Angell. Industry-sponsored clinical research: a broken system. *Jama*, 300 (9):1069–1071, 2008.

[41] Deborah Cohen. FDA official "clinical trial system is broken". *BMJ*, 347:f6980, 2013.

[42] Trisha Greenhalgh, Jeremy Howick, and Neal Maskrey. Evidence based medicine: a movement in crisis? *BMJ*, 348:g3725, June 2014. ISSN 1756-1833. doi: 10.1136/bmj.g3725.

[43] Syed Muhammad Umar Nasser, Graham Cooke, Katharina Kranzer, Susan L. Norris, Piero Olliaro, and Nathan Ford. Strength of recommendations in WHO guidelines using GRADE was associated with uptake in national policy. *Journal of Clinical Epidemiology*, 68(6):703–707, June 2015. ISSN 0895-4356. doi: 10.1016/j.jclinepi.2014.11.006.

[44] Steve Kanters, Nathan Ford, Eric Druyts, Kristian Thorlund, Edward J. Mills, and Nick Bansback. Use of network meta-analysis in clinical guidelines. *Bulletin of the World Health Organization*, 94(10):782, 2016.

[45] D. Allen and K. J. Harkins. Too much guidance? *The Lancet*, 365(9473):1768, 2005.

[46] Kathleen N. Lohr, Marilyn J. Field, and others. *Clinical practice guidelines: directions for a new program*, volume 90. National Academies Press, 1990.

[47] Institute of Medicine (US) Committee on Standards for Developing Trustworthy Clinical Practice Guidelines. *Clinical Practice Guidelines We Can Trust*. National Academies Press (US), Washington (DC), 2011. ISBN 978-0-309-16422-1 978-0-309-16423-8.

[48] Marilyn J. Field, Kathleen N. Lohr, and others. *Guidelines for Clinical Practice:: From Development to Use*. National Academies Press, 1992.

[49] Arturo González-Ferrer, Annette ten Teije, Juan Fernández-Olivares, and Krystyna Milian. Automated generation of patient-tailored electronic care pathways by translating computer-interpretable guidelines into hierarchical task networks. *Artificial Intelligence in Medicine*, 57(2):91–109, 2013.

[50] Enrico Coiera. *Guide to health informatics*. CRC press, 2015.

[51] Richard N. Shiffman, George Michel, Richard M. Rosenfeld, and Caryn Davidson. Building better guidelines with BRIDGE-Wiz: development and evaluation of a software assistant to promote clarity, transparency and implementability. *Journal of American Medical Informatics Association*, 19:94–101, 2012.

[52] Steven H. Woolf, Richard Grol, Allen Hutchinson, Martin Eccles, and Jeremy Grimshaw. Potential benefits, limitations and harms of clinical guidelines. *BMJ*, 318:527–530, 1999.

[53] David Isern and Antonio Moreno. Computer-based execution of clinical guidelines: A review. *International Journal of Medical Informatics*, 77(12):787–808, 2008.

[54] Mike English and Newton Opiyo. Getting to grips with GRADE—perspective from a low-income setting. *Journal of clinical epidemiology*, 64(7):708–710, July 2011. ISSN 0895-4356. doi: 10.1016/j.jclinepi.2010.07.016.

[55] Rudolph Wabitsch, Carmi Z. Margolis, Jean-Elie Malkin, Kathleen Abu-Saad, and Javier Waksman. Evaluating the process of tailoring the global HIV/AIDS practice guidelines for use in developing countries. *International Journal for Quality in Health Care*, 10(2):147–154, April 1998. ISSN 1353-4505. doi: 10.1093/intqhc/10.2.147.

[56] Charles F Gilks, Siobhan Crowley, René Ekpini, Sandy Gove, Jos Perriens, Yves Souteyrand, Don Sutherland, Marco Vitoria, Teguest Guerma, and Kevin De Cock. The WHO public-health approach to antiretroviral treatment against HIV in resource-limited settings. *The Lancet*, 368(9534):505–510, August 2006. ISSN 0140-6736. doi: 10.1016/S0140-6736(06)69158-7.

[57] Rochelle P. Walensky, Robin Wood, Andrea L. Ciaranello, A. David Paltiel, Sarah B. Lorenzana, Xavier Anglaret, Adam W. Stoler, Kenneth A. Freedberg, and for the CEPAC-International Investigators. Scaling Up the 2010 World Health Organization HIV Treatment Guidelines in Resource-Limited Settings: A Model-Based Analysis. *PLOS Medicine*, 7(12):e1000382, December 2010. ISSN 1549-1676. doi: 10.1371/journal.pmed.1000382.

[58] Motasim Badri, Linda-Gail Bekker, Catherine Orrell, Jennifer Pitt, François Cilliers, and Robin Wood. Initiating highly active antiretroviral therapy in sub-Saharan Africa: an assessment of the revised World Health Organization scaling-up guidelines. *AIDS*, 18(8):1159–1168, 2004.

[59] Sri Rezeki S. Hadinegoro. The revised WHO dengue case classification: does the system need to be modified? *Paediatrics and international child health*, 32(sup1): 33–38, 2012.

[60] Mike English and J. Anthony G. Scott. What Is the Future for Global Case Management Guidelines for Common Childhood Diseases? *PLOS Medicine*, 5 (12):e241, December 2008. ISSN 1549-1676. doi: 10.1371/journal.pmed.0050241.

[61] Katharina Kaiser and Silvia Miksch. Versioning computer-interpretable guidelines: Semi-automatic modeling of 'living guidelines' using an information extraction method. *Artificial Intelligence in Medicine*, 46(1):55–66, 2009. ISSN 0933-3657. doi: 10.1016/j.artmed.2008.08.009.

[62] Annette ten Teije, Mar Marcos, Michel Balser, Joyce van Croonenborg, Christoph Duelli, Frank van Harmelen, Peter Lucas, Silvia Miksch, Wolfgang Reif, Kitty Rosenbrand, and others. Improving medical protocols by formal methods. *Artificial intelligence in medicine*, 36(3):193–209, 2006.

[63] Alessio Bottrighi, Gianpaolo Molino, Stefania Montani, Paolo Terenziani, and Mauro Torchio. Supporting a distributed execution of clinical guidelines. *Computer Methods and Programs in Biomedicine*, 112(1):200–210, 2013.

[64] Samson W. Tu, Mark A. Musen, and others. Modeling data and knowledge in the EON guideline architecture. *Studies in health technology and informatics*, (1): 280–284, 2001.

[65] Allan T. Pryor and George Hripcsak. Sharing MLM's: an experiment between Columbia-Presbyterian and LDS Hospital. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 399. American Medical Informatics Association, 1993.

[66] Samson W. Tu, James Campbell, and Mark A. Musen. The SAGE guideline modeling: motivation and methodology. *Studies in health technology and informatics*, pages 167–171, 2004.

[67] Mor Peleg. Guidelines and workflow models. *Clinical Decision Support-The Road Ahead*, pages 281–306, 2007.

[68] Filipe Gonçalves, Tiago Oliveira, José Neves, and Paulo Novais. CompGuide: Acquisition and Editing of Computer-Interpretable Guidelines. In *Recent Advances in Information Systems and Technologies*, pages 257–266. Springer, Cham, April 2017. doi: 10.1007/978-3-319-56535-4_26.

[69] Nassim Douali, Huszka Csaba, Jos De Roo, Elpiniki I. Papageorgiou, and Marie-Christine Jaulent. Diagnosis Support System based on clinical guidelines: comparison between Case-Based Fuzzy Cognitive Maps and Bayesian Networks. *Computer Methods and Programs in Biomedicine*, 113(1):133–143, 2014.

[70] Greg C. Hagerty, David Pickens, Casimir Kulikowski, and Frank Sonnenberg. HGML: a hypertext guideline markup language. In *Proceedings of the AMIA Symposium*, page 325. American Medical Informatics Association, 2000.

[71] Richard N. Shiffman, Bryant T. Karras, Abha Agrawal, Roland Chen, Luis Marenco, and Sujai Nath. GEM: a proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Association*, 7(5):488–498, 2000.

[72] George Hripcsak, Paul D. Clayton, Allan T. Pryor, P. Haug, O. B. Wigertz, and J. Van der Lei. The Arden Syntax for Medical Logic Modules. In *Proceedings. Symposium on Computer Applications in Medical Care*, pages 200–204. American Medical Informatics Association, 1990.

[73] Health Level Seven International. The arden syntax for medical logic systems version 2.9. Technical report, Health Level Seven International, 2013.

[74] George Hripcsak, Ove B. Wigertz, and Paul D. Clayton. Origins of the Arden syntax. *Artificial intelligence in medicine*, 2015.

[75] Mor Peleg, Aziz A. Boxwala, Omolola Ogunyemi, Qing Zeng, Samson Tu, Ronilda Lacson, Elmer Bernstam, Nachman Ash, Peter Mork, Lucila Ohno-Machado, et al. Glif3: the evolution of a guideline representation format. page 645, 2000.

[76] Silvia Miksch, Yuval Shahar, and Peter Johnson. Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans. In *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*, pages 9–19. Milton Keynes, UK, The Open University, Milton Keynes, UK, 1997.

[77] David R. Sutton and John Fox. The Syntax and Semantics of the PROforma Guideline Modeling Language. *Journal of the American Medical Informatics Association*, 10(5):433–443, September 2003. ISSN , 1527-974X. doi: 10.1197/jamia.M1264.

[78] F. A. Sonnenberg and C. G. Hagerty. Computer-interpretable clinical practice guidelines. Where are we and where are we going ? *Yearbook of Medical Informatics*, pages 145–158, 2006. ISSN 0943-4747.

[79] Kai Zheng, Rema Padman, Michael P. Johnson, and Sharique Hasan. Guideline representation ontologies for evidence-based medicine practice. *Handbook of Research on Advances in Health Informatics and Electronic Healthcare Applications: Global Adoption and Impact of Information Communication Technologies*, page 234, 2009.

[80] Samson W. Tu, James R. Campbell, Julie Glasgow, Mark A. Nyman, Robert McClure, James McClay, Craig Parker, Karen M. Hrabak, David Berg, Tony Weida, James G. Mansfield, Mark A. Musen, and Robert M. Abarbanel. The SAGE Guideline Model: Achievements and Overview. *Journal of the American Medical Informatics Association : JAMIA*, 14(5):589–598, 2007. ISSN 1067-5027. doi: 10.1197/jamia.M2399.

[81] George Hripcsak. Writing Arden Syntax medical logic modules. *Computers in Biology and Medicine*, 24(5):331–363, September 1994. ISSN 0010-4825. doi: 10.1016/0010-4825(94)90002-7.

[82] Stefan Kraus, Ixchel Castellanos, Dennis Toddenroth, Hans-Ulrich Prokosch, and Thomas Bürkle. Integrating Arden-Syntax-based clinical decision support with extended presentation formats into a commercial patient data management system. *Journal of Clinical Monitoring and Computing*, 28(5):465–473, October 2014. ISSN 1387-1307, 1573-2614. doi: 10.1007/s10877-013-9430-0.

[83] Alexander Seitinger, Andrea Rappelsberger, Harald Leitich, Michael Binder, and Klaus-Peter Adlassnig. Executable medical guidelines with Arden Syntax—Applications in dermatology and obstetrics. *Artificial Intelligence in Medicine*, August 2016. ISSN 0933-3657. doi: 10.1016/j.artmed.2016.08.003.

[84] Matthias Samwald, Karsten Fehre, Jeroen de Bruin, and Klaus-Peter Adlassnig. The Arden Syntax standard for clinical decision support: Experiences and directions. *Journal of Biomedical Informatics*, 45(4):711–718, August 2012. ISSN 1532-0464. doi: 10.1016/j.jbi.2012.02.001.

[85] Samson W. Tu and Mark A. Musen. The EON model of intervention protocols and guidelines. In *Proceedings of the AMIA Annual Fall Symposium*, page 587. American Medical Informatics Association, 1996.

[86] Mark A. Musen, Samson W. Tu, Amar Das, and Yuval Shahar. EON: a component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association*, 3(6), 1996.

[87] Samson W. Tu and Mark A. Musen. From guideline modeling to guideline execution: defining guideline-based decision-support services. *Proceedings of the AMIA Symposium*, pages 863–867, 2000. ISSN 1531-605X.

[88] Mor Peleg, Samson Tu, Jonathan Bury, Paolo Ciccarese, John Fox, Robert A. Greenes, Richard Hall, Peter D. Johnson, Neill Jones, Anand Kumar, Silvia Miksch, Silvana Quaglini, Andreas Seyfang, Edward H. Shortliffe, and Mario Stefanelli. Comparing Computer-interpretable Guideline Models: A Case-study Approach. *Journal of the American Medical Informatics Association : JAMIA*, 10 (1):52–68, 2003. ISSN 1067-5027. doi: 10.1197/jamia.M1135.

[89] Peter D. Johnson, Samson Tu, Nick Booth, Bob Sugden, and Ian N. Purves. Using scenarios in chronic disease management guidelines for primary care. page 389, 2000.

[90] Samson W. Tu, Karen M. Hrabak, James R. Campbell, Julie Glasgow, Mark A. Nyman, Robert McClure, James McClay, Robert Abarbanel, James G. Mansfield, Susana M. Martins, Mary K. Goldstein, and Mark A. Musen. Use of Declarative Statements in Creating and Maintaining Computer-Interpretable Knowledge Bases for Guideline-Based Care. *AMIA Annual Symposium Proceedings*, 2006:784–788, 2006. ISSN 1942-597X.

[91] David R. Sutton, Paul Taylor, and Kenneth Earle. Evaluation of PROforma as a language for implementing medical guidelines in a practical context. *BMC Medical Informatics and Decision Making*, 6(1):20, April 2006. ISSN 1472-6947. doi: 10.1186/1472-6947-6-20.

[92] Silvana Quaglini, Mario Stefanelli, Giordano Lanzola, Vincenzo Caporusso, and Silvia Panzarasa. Flexible guideline-based patient careflow systems. *Artificial Intelligence in Medicine*, 22(1):65–80, April 2001. ISSN 0933-3657. doi: 10.1016/S0933-3657(00)00100-7.

[93] Silvana Quaglini, Mario Stefanelli, Anna Cavallini, Giuseppe Micieli, C. Fassino, and C. Mossa. Guideline-based careflow systems. *Artificial Intelligence in Medicine*, 20(1):5–22, September 2000. ISSN 0933-3657. doi: 10.1016/S0933-3657(00)00050-6.

[94] Vimla L. Patel, Vanessa G. Allen, José F. Arocha, and Edward H. Shortliffe. Representing Clinical Guidelines in GLIF. *Journal of the American Medical Informatics Association*, 5(5):467–483, September 1998. ISSN 1067-5027, 1527-974X. doi: 10.1136/jamia.1998.0050467.

[95] Aziz A. Boxwala, Mor Peleg, Samson Tu, Omolola Ogunyemi, Qing T. Zeng, Dongwen Wang, Vimla L. Patel, Robert A. Greenes, and Edward H. Shortliffe. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. *Journal of Biomedical Informatics*, 37(3):147–161, June 2004. ISSN 1532-0464. doi: 10.1016/j.jbi.2004.04.002.

[96] Margarita Sordo, Omolola Ogunyemi, Aziz A. Boxwala, and Robert A. Greenes. GELLO: An Object-Oriented Query and Expression Language for Clinical Decision Support. *AMIA Annual Symposium Proceedings*, 2003:1012, 2003. ISSN 1942-597X.

[97] Dongwen Wang, Mor Peleg, Samson W. Tu, Aziz A. Boxwala, Robert A. Greenes, Vimla L. Patel, and Edward H. Shortliffe. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines:: A literature review of guideline representation models. *International Journal of Medical Informatics*, 68(1):59–70, 2002. ISSN 1386-5056. doi: 10.1016/S1386-5056(02) 00065-5.

[98] Paul A. de Clercq, Arie Hasman, Johannes A. Blom, and Hendrikus H. M. Korsten. Design and implementation of a framework to support the development of clinical guidelines. *International Journal of Medical Informatics*, 64(2–3):285–318, December 2001. ISSN 1386-5056. doi: 10.1016/S1386-5056(01)00189-7.

[99] Yuval Shahar, Silvia Miksch, and Peter Johnson. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1–2):29–51, September 1998. ISSN 0933-3657. doi: 10.1016/S0933-3657(98)00015-3.

[100] Isabelle Colombet, Angel-Ricardo Aguirre-Junco, Sylvain Zunino, Marie-Christine Jaulent, Laurence Leneveut, and Gilles Chatellier. Electronic implementation of guidelines in the EsPeR system: a knowledge specification method. *International journal of medical informatics*, 74(7):597–604, 2005.

[101] Brigitte Seroussi, Jacques Bouaud, and G. Chatellier. Guideline-based modeling of therapeutic strategies in the special case of chronic diseases. *International journal of medical informatics*, 74(2):89–99, 2005.

[102] Beatriz Pérez and Ivan Porres. Authoring and verification of clinical guidelines: A model driven approach. *Journal of Biomedical Informatics*, 43(4):520–536, August 2010. ISSN 1532-0464. doi: 10.1016/j.jbi.2010.02.009.

[103] Bernard Kamsu-Foguem, Germaine Tchuenté-Foguem, and Clovis Foguem. Using conceptual graphs for clinical guidelines representation and knowledge visualization. *Information Systems Frontiers*, 16(4):571–589, September 2014. ISSN 1387-3326, 1572-9419. doi: 10.1007/s10796-012-9360-2.

[104] David Isern, David Sánchez, and Antonio Moreno. Ontology-driven execution of clinical guidelines. *Computer Methods and Programs in Biomedicine*, 107(2): 122–139, 2012.

[105] David Riano, Francis Real, Joan Albert López-Vallverdú, Fabio Campana, Sara Ercolani, Patrizia Mecocci, Roberta Annicchiarico, and Carlo Caltagirone. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. *Journal of Biomedical Informatics*, 45(3):429–446, June 2012. ISSN 1532-0464. doi: 10.1016/j.jbi.2011.12.008.

[106] Samina Abidi, Jafna Cox, S. Abidi, and Michael Shepherd. Using OWL ontologies for clinical guidelines based comorbid decision support. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 3030–3038. IEEE, 2012.

[107] Zhisheng Huang, Annette ten Teije, Frank van Harmelen, and Salah Aït-Mokhtar. Semantic Representation of Evidence-Based Clinical Guidelines. In Silvia Miksch, David Riaño, and Annette ten Teije, editors, *Knowledge Representation for Health Care*, number 8903 in Lecture Notes in Computer Science, pages 78–94. Springer International Publishing, July 2014. ISBN 978-3-319-13280-8 978-3-319-13281-5.

[108] Veruska Zamborlini, Rinke Hoekstra, Marcos da Silveira, Cédric Pruski, Annette ten Teije, and Frank van Harmelen. A conceptual model for detecting interactions among medical recommendations in clinical guidelines. In *Knowledge Engineering and Knowledge Management*, pages 591–606. Springer, 2014.

[109] Rinke Hoekstra, A. Waard, Richard Vdovjak, and others. Annotating Evidence Based Clinical Guidelines: A Lightweight Ontology. In *Proceedings of the 5th International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2012)*, volume 952. CEUR, 2012.

[110] Barend Mons, Herman van Haagen, Christine Chichester, Peter-Bram 't Hoen, Johan T. den Dunnen, Gertjan van Ommen, Erik van Mulligen, Bharat Singh, Rob Hooft, Marco Roos, Joel Hammond, Bruce Kiesel, Belinda Giardine, Jan Velterop, Paul Groth, and Erik Schultes. The value of data. *Nature Genetics*, 43 (4):281–283, April 2011. ISSN 1061-4036. doi: 10.1038/ng0411-281.

[111] Katharina Kaiser and Mar Marcos. Leveraging workflow control patterns in the domain of clinical practice guidelines. *BMC medical informatics and decision making*, 16(1):20, 2016.

[112] Salvador Rodriguez Loya, Ayesha Aziz, and Chris Chatwin. A service oriented approach for guidelines-based clinical decision support using BPMN. *Studies in health technology and informatics*, 205, 2014.

[113] Richard Braun, Hannes Schlieter, Martin Burwitz, and Werner Esswein. BPMN4cp: Design and implementation of a BPMN extension for clinical pathways. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 9–16. IEEE, 2014.

[114] Ayesha Aziz, Salvador Rodriguez, and Chris Chatwin. From guidelines to practice: improving clinical care through rule-based clinical decision support at the point of care. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 178–185. Springer, 2014.

[115] Perry L. Miller, Sandra J. Frawley, and Frederick G. Sayward. Maintaining and Incrementally Revalidating a Computer-Based Clinical Guideline: A Case Study. *Journal of Biomedical Informatics*, 34(2):99–111, April 2001. ISSN 1532-0464. doi: 10.1006/jbin.2001.1011.

[116] Mor Peleg, Samson Tu, and others. Decision support, knowledge representation and management in medicine. *Yearb Med Inform*, 45:72–80, 2006.

[117] Fabio Grandi, Federica Mandreoli, and Riccardo Martoglia. Efficient management of multi-version clinical guidelines. *Journal of Biomedical Informatics*, 45(6):1120–1136, December 2012. ISSN 1532-0464. doi: 10.1016/j.jbi.2012.07.005.

[118] Mor Peleg and Rami Kantor. Approaches for Guideline Versioning Using GLIF. *AMIA Annual Symposium Proceedings*, 2003:509–513, 2003. ISSN 1942-597X.

[119] Yuval Shahar, Ohad Young, Erez Shalom, Alon Mayaffit, Robert Moskovitch, Alon Hessing, and Maya Galperin. DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines. *Artificial Intelligence in Medicine*, pages 122–131, 2003.

[120] Brian DeRenzi, Neal Lesh, Tapan Parikh, Clayton Sims, Werner Maokla, Mwajuma Chemba, Yuna Hamisi, David S hellenberg, Marc Mitchell, and Gaetano Borriello. E-imci: Improving Pediatric Health Care in Low-income Countries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 753–762, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357174.

[121] Joaquin A. Blaya, Hamish S. F. Fraser, and Brian Holt. E-Health Technologies Show Promise In Developing Countries. *Health Affairs*, 29(2):244–251, February 2010. ISSN 0278-2715, 1544-5208. doi: 10.1377/hlthaff.2009.0894.

[122] Gerald P. Douglas, Oliver J. Gadabu, Sabine Joukes, Soyapi Mumba, Michael V. McKay, Anne Ben-Smith, Andreas Jahn, Erik J. Schouten, Zach Landis Lewis, Joep J. van Oosterhout, and others. Using touchscreen electronic medical record systems to support and monitor national scale-up of antiretroviral therapy in Malawi. *PLoS medicine*, 7(8):e1000319, 2010.

[123] Moya Conrick. *Health Informatics*. Cengage Learning Australia, 2006.

[124] Edward H. Shortliffe and James J. Cimino. *Biomedical informatics: computer applications in health care and biomedicine*. Springer Science & Business Media, 2013.

[125] Ramona Nelson and Nancy Staggers. *Health Informatics-E-Book: An Interprofessional Approach*. Elsevier Health Sciences, 2016.

[126] Dave Garets and Mike Davis. Electronic medical records vs. electronic health records: yes, there is a difference. *Policy white paper. Chicago, HIMSS Analytics*, pages 1–14, 2006.

[127] Tsipi Heart, Ofir Ben-Assuli, and Itamar Shabtai. A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy. *Health Policy and Technology*, 6(1):20–25, 2017.

[128] Nina Boulus and Pernille Bjorn. A cross-case analysis of technology-in-use practices: EPR-adaptation in Canada and Norway. *international journal of medical informatics*, 79(6):e97–e108, 2010.

[129] Hamish S. F. Fraser and Joaquin Blaya. Implementing medical information systems in developing countries, what works and what doesn't. In *AMIA Annual Symposium Proceedings*, volume 2010, page 232. American Medical Informatics Association, 2010.

[130] Peter Howitt, Ara Darzi, Guang-Zhong Yang, Hutan Ashrafian, Rifat Atun, James Barlow, Alex Blakemore, Anthony MJ Bull, Josip Car, and Lesong Conteh. Technologies for global health. *The Lancet*, 380(9840):507–535, 2012.

[131] Lameck Diero, Joseph K. Rotich, John Bii, Burke W. Mamlin, Robert M. Einterz, Irene Z. Kalamai, and William M. Tierney. A computer-based medical record system and personal digital assistants to assess and follow patients with respiratory

tract infections visiting a rural Kenyan health centre. *BMC Medical Informatics and Decision Making*, 6(1):21, 2006.

[132] Hamish S. F. Fraser, Christian Allen, Christopher Bailey, Gerry Douglas, Sonya Shin, and Joaquin Blaya. Information systems for patient follow-up and chronic management of HIV and tuberculosis: a life-saving technology in resource-poor areas. *Journal of medical Internet research*, 9(4), 2007.

[133] GB Kouematchoua Tchuitcheu and O. Rienhoff. Options for diabetes management in sub-Saharan Africa with an electronic medical record system. *Methods of information in medicine*, 50(1):11–22, 2011.

[134] Caroline Abdul Ngoma, Honest Christopher Kimaro, Faraja Teddy Igira, John Mukulu, Jens Johan Kaasbøll, Alice Yona, Naza Mkabara, Siaeli Sarikoki, Immaculata Sangu, Cecilia Njau, and others. Challenges in Implementing a Computerized Name-based Information Tracking System: Practical Experiences from Maternal Health Care. *IST-Africa 2012 Conference Proceedings*, 2012.

[135] Benjamin H. Chi, Bellington Vwalika, William P. Killam, Chibesa Wamalume, Mark J. Giganti, Reuben Mbewe, Elizabeth M. Stringer, Namwinga T. Chintu, Nande B. Putta, Katherine C. Liu, Carla J. Chibwesha, Dwight J. Rouse, and Jeffrey S. A. Stringer. Implementation of the Zambia Electronic Perinatal Record System for comprehensive prenatal and delivery care. *International Journal of Gynecology & Obstetrics*, 113(2):131–136, May 2011. ISSN 0020-7292. doi: 10.1016/j.ijgo.2010.11.013.

[136] Marlen S. Chawani. A Cross-case Analysis of the Effects of EMR Deployment on Antenatal Care Services in Rural Health Centres in Malawi. *Journal of Health Informatics in Africa*, 2(1), December 2014. ISSN 2197-6902.

[137] Joseph K. Rotich, Terry J. Hannan, Faye E. Smith, John Bii, Wilson W. Odero, Nguyen Vu, Burke W. Mamlin, Joseph J. Mamlin, Robert M. Einterz, and William M. Tierney. Installing and implementing a computer-based patient record system in sub-Saharan Africa: the Mosoriot Medical Record System. *Journal of the American Medical Informatics Association*, 10(4):295–303, 2003.

[138] Evan Waters, Jeff Rafter, Gerald P. Douglas, Mwatha Bwanali, Darius Jazayeri, and H. S. Fraser. Experience implementing a point-of-care electronic medical record system for primary care in Malawi. *Studies in health technology and informatics*, 160(Pt 1):96–100, 2010.

[139] Christopher J. Seebregts, Burke W. Mamlin, Paul G. Biondich, Hamish S. F. Fraser, Benjamin A. Wolfe, Darius Jazayeri, Christian Allen, Justin Miranda,

Elaine Baker, and Nicholas Musinguzi. The OpenMRS implementers network. *International journal of medical informatics*, 78(11):711–720, 2009.

[140] Maurice Mars. Building the capacity to build capacity in e-health in sub-Saharan Africa: the KwaZulu-Natal experience. *Telemedicine and e-Health*, 18(1):32–37, 2012.

[141] Peter S. Millard, Juan Bru, and Christopher A. Berger. Open-source point-of-care electronic medical records for use in resource-limited settings: systematic review and questionnaire surveys. *BMJ Open*, 2(4):e000690, January 2012. ISSN 2044-6055,. doi: 10.1136/bmjopen-2011-000690.

[142] Miss Laiha Mat Kiah, Ahmed Haiqi, BB Zaidan, and AA Zaidan. Open source emr software: profiling, insights and hands-on analysis. *Computer methods and programs in biomedicine*, 117(2):360–382, 2014.

[143] William Hersh, Alvaro Margolis, Fernán Quirós, and Paula Otero. Building a health informatics workforce in developing countries. *Health Affairs*, 29(2):274–277, 2010.

[144] Gloria Ser, Ann Robertson, and Aziz Sheikh. A qualitative exploration of workarounds related to the implementation of national electronic health records in early adopter mental health hospitals. *PloS one*, 9(1):e77669, 2014.

[145] Karen E. Joynt, Deepak L. Bhatt, Lee H. Schwamm, Ying Xian, Paul A. Heidenreich, Gregg C. Fonarow, Eric E. Smith, Megan L. Neely, Maria V. Grau-Sepulveda, and Adrian F. Hernandez. Lack of impact of electronic health records on quality of care and outcomes for ischemic stroke. *Journal of the American College of Cardiology*, 65(18):1964–1972, 2015.

[146] Burke W. Mamlin, Paul G. Biondich, Ben A. Wolfe, Hamish Fraser, Darius Jazayeri, Christian Allen, Justin Miranda, and William M. Tierney. Cooking up an open source EMR for developing countries: OpenMRS–a recipe for successful collaboration. In *AMIA Annual Symposium Proceedings*, volume 2006, page 529. American Medical Informatics Association, 2006.

[147] Martin C. Were, Changyu Shen, Mwebesa Bwana, Nneka Emenyonu, Nicholas Musinguzi, Frank Nkuyahaga, Annet Kembabazi, and William M. Tierney. Creation and evaluation of EMR-based paper clinical summaries to support HIV-care in Uganda, Africa. *International journal of medical informatics*, 79(2):90–96, 2010.

[148] Yaw Anokwa, Nyoman Ribeka, Tapan Parikh, Gaetano Borriello, and Martin C. Were. Design of a phone-based clinical decision support system for resource-limited

settings. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, pages 13–24. ACM, 2012.

[149] Darren Desantis, Richard J. Baverstock, Andrea Civitarese, R. Trafford Crump, and Kevin V. Carlson. A clinical perspective on electronically collecting patient-reported outcomes at the point-of-care for overactive bladder. *Canadian Urological Association Journal*, 10(11-12):E359, 2016.

[150] Barbara Castelnuovo, Agnes Kiragga, Victor Afayo, Malisa Ncube, Richard Orama, Stephen Magero, Peter Okwi, Yukari C. Manabe, and Andrew Kambugu. Implementation of provider-based electronic medical records and improvement of the quality of data in a large HIV program in Sub-Saharan Africa. *PloS one*, 7 (12):e51631, 2012.

[151] Patrick J. O'Connor, Jay R. Desai, John C. Butler, Elyse O. Kharbanda, and JoAnn M. Sperl-Hillen. Current status and future prospects for electronic point-of-care clinical decision support in diabetes care. *Current diabetes reports*, 13(2): 172–176, 2013.

[152] Robert France and Bernhard Rumpe. Model-driven Development of Complex Software: A Research Roadmap. In *2007 Future of Software Engineering*, FOSE '07, pages 37–54, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2829-5. doi: 10.1109/FOSE.2007.14.

[153] Markus Voelter, Sebastian Benz, Christian Dietrich, Birgit Engelmann, Mats Helander, Lennart CL Kats, Eelco Visser, and Guido Wachsmuth. *DSL engineering: Designing, implementing and using domain-specific languages.* dslbook. org, 2013.

[154] Wei Ke, Xiaoshan Li, Zhiming Liu, and Volker Stolz. rCOS: a formal model-driven engineering method for component-based software. *Frontiers of Computer Science*, 6(1):17–39, January 2012. ISSN 1673-7350, 1673-7466. doi: 10.1007/ s11704-012-2901-5.

[155] Michael Heß, Monika Kaczmarek, Ulrich Frank, Lars Podleska, and Georg Täger. A Domain-specific Modelling Language for Clinical Pathways in the Realm of Multi-perspective Hospital Modelling. In *ECIS*, 2015.

[156] Richard Braun, Hannes Schlieter, Martin Burwitz, and Werner Esswein. Extending a Business Process Modeling Language for Domain-Specific Adaptation in Healthcare. In *Wirtschaftsinformatik*, pages 468–481, 2015.

[157] Martin Fowler. *Domain-Specific Languages.* Pearson Education, September 2010. ISBN 978-0-13-139280-9.

[158] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and How to Develop Domain-specific Languages. *ACM Comput. Surv.*, 37(4):316–344, December 2005. ISSN 0360-0300. doi: 10.1145/1118890.1118892.

[159] Jennifer Munnelly and Siobhán Clarke. A domain-specific language for ubiquitous healthcare. In *Pervasive Computing and Applications, 2008. ICPCA 2008. Third International Conference on*, volume 2, pages 757–762. IEEE, 2008.

[160] Jean E. Sammet. *Programming Languages: History and Fundamentals.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1969. ISBN 0-13-729988-5.

[161] Richard L. Wexelblat. *History of programming languages.* Academic Press, 1981.

[162] James. Martin. *Fourth-Generation Languages. Vol. I: Principles, Vol II: Representative 4GLs.* Prentice-Hall, 1985.

[163] Jon Bentley. Programming pearls: little languages. *Communications of the ACM*, 29(8):711–721, 1986.

[164] Bonnie A. Nardi. *A small matter of programming: perspectives on end user computing.* MIT press, 1993.

[165] Thomas J. Bergin and Richard G. Gibson. *History of programming languages II.* ACM Press New York, 1996.

[166] Jonathan Sprinkle, Marjan Mernik, Juha-Pekka Tolvanen, and Diomidis Spinellis. Guest Editors' Introduction: What Kinds of Nails Need a Domain-Specific Hammer? *IEEE Software*, 26(4):15–18, July 2009. ISSN 0740-7459. doi: 10.1109/MS.2009.92.

[167] Mark Strembeck and Uwe Zdun. An approach for the systematic development of domain-specific languages. *Software: Practice and Experience*, 39(15):1253–1292, 2009.

[168] Scott A. Thibault, Renaud Marlet, and Charles Consel. Domain-specific languages: From design to implementation application to video device drivers generation. *IEEE Transactions on software Engineering*, 25(3):363–377, 1999.

[169] David Wile. Lessons learned from real DSL experiments. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003.

[170] Janne Luoma, Steven Kelly, and Juha-Pekka Tolvanen. Defining domain-specific modeling languages: Collected experiences. In *4th Workshop on Domain-Specific Modeling*, 2004.

[171] Jack Greenfield and Keith Short. Software factories: assembling applications with patterns, models, frameworks and tools. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 16–27. ACM, 2003.

[172] Thomas Stahl and Markus Voelter. *Model-driven software development*. J. Wiley & Sons, 2006.

[173] Jesús Sánchez Cuadrado and Jesús García Molina. Building domain-specific languages for model-driven development. *IEEE software*, 24(5), 2007.

[174] Steven Kelly and Juha-Pekka Tolvanen. *Domain-specific modeling: enabling full code generation*. John Wiley & Sons, 2008.

[175] Paul Hudak. Building domain-specific embedded languages. *ACM Computing Surveys (CSUR)*, 28(4es):196, 1996.

[176] Diomidis Spinellis. Notable design patterns for domain-specific languages. *Journal of systems and software*, 56(1):91–99, 2001.

[177] E. Grant, Krish Narayanan, and Hassan Reza. Rigorously defined domain modeling languages. In *Proc. of the 4th OOPSLA Workshop on Domain-Specific Modeling*, 2004.

[178] Ryan Paul. Designing and implementing a domain-specific language. *Linux Journal*, 2005(135):7, 2005.

[179] Tomaž Kosar, Pablo E. Martinez Lopez, Pablo A. Barrientos, and Marjan Mernik. A preliminary study on various implementation approaches of domain-specific language. *Information and software technology*, 50(5):390–405, 2008.

[180] Gabor Karsai, Holger Krahn, Claas Pinkernell, Bernhard Rumpe, Martin Schindler, and Steven Völkel. Design guidelines for domain specific languages. In *9th OOPSLA Workshop on Domain-Specific Modeling (DSM'09)*, pages 7–13, 2009.

[181] Anneke Kleppe. *Software language engineering: creating domain-specific languages using metamodels*. Pearson Education, 2008.

[182] Arie Van Deursen, Paul Klint, and Joost Visser. Domain-Specific Languages: An Annotated Bibliography. *Sigplan Notices*, 35(6):26–36, 2000.

[183] Elvinia Riccobene, Paolo Arcaini, Patrizia Scandurra, and Angelo Gargantini. Formal semantics for metamodel-based domain specific languages. *Computational Linguistics: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*, page 297, 2014.

[184] Luca Cardelli. Type systems. *ACM Computing Surveys*, 28(1):263–264, 1996.

[185] István Ráth, András Ökrös, and Dániel Varró. Synchronization of abstract and concrete syntax in domain-specific modeling languages. *Software & Systems Modeling*, 9(4):453–471, August 2009. ISSN 1619-1366, 1619-1374. doi: 10.1007/s10270-009-0122-7.

[186] John C. Reynolds. *Theories of Programming Languages*. Cambridge University Press, April 2009. ISBN 978-0-521-10697-9. Google-Books-ID: 2OwlTC4SOccC.

[187] Robert Harper. *Practical foundations for programming languages*. Cambridge University Press, 2016.

[188] Raymond Turner. *Computable Models*. Springer Publishing Company, Incorporated, 1 edition, 2009. ISBN 1-84882-051-8 978-1-84882-051-7.

[189] Fabien Campagne. *The MPS Language Workbench: Volume I*, volume 1. Fabien Campagne, 2014.

[190] Naveneetha Vasudevan and Laurence Tratt. Comparative Study of DSL Tools. *Electronic Notes in Theoretical Computer Science*, 264(5):103–121, July 2011. ISSN 1571-0661. doi: 10.1016/j.entcs.2011.06.007.

[191] Sebastian Erdweg, Tijs Van Der Storm, Markus Völter, Laurence Tratt, Remi Bosman, William R Cook, Albert Gerritsen, Angelo Hulshout, Steven Kelly, Alex Loh, et al. Evaluating and comparing language workbenches: Existing results and benchmarks for the future. volume 44, pages 24–47. Elsevier, 2015.

[192] Roman Stoffel. Comparing Language Workbenches. In *MSE-seminar: Program Analysis and Transformation, University of Applied Sciences Rapperswil (HSR), Switzerland*, pages 18–24, 2010.

[193] André Ribeiro, Luís de Sousa, and Alberto Rodrigues da Silva. Comparative Analysis of Workbenches to Support DSMLs: Discussion with Non-Trivial Model-Driven Development Needs. In *MODELSWARD*, pages 323–330, 2016.

[194] Kari Smolander, Kalle Lyytinen, Veli-Pekka Tahvanainen, and Pentti Marttiin. MetaEdit—a flexible graphical environment for methodology modelling. In *Advanced Information Systems Engineering*, pages 168–193. Springer, 1991.

[195] Steven Kelly, Kalle Lyytinen, and Matti Rossi. Metaedit+ a fully configurable multi-user and multi-tool case and came environment. In *Advanced Information Systems Engineering*, pages 1–21. Springer, 1996.

[196] Juha-Pekka Tolvanen and Matti Rossi. MetaEdit+: defining and using domain-specific modeling languages and code generators. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 92–93. ACM, 2003.

[197] Steven Kelly, Kalle Lyytinen, Matti Rossi, and Juha Pekka Tolvanen. MetaEdit+ at the Age of 20. In *Seminal Contributions to Information Systems Engineering*, pages 131–137. Springer, 2013.

[198] Louis M. Rose, Dimitrios S. Kolovos, and Richard F. Paige. Eugenia live: a flexible graphical modelling tool. In *Proceedings of the 2012 Extreme Modeling Workshop*, pages 15–20. ACM, 2012.

[199] Dave Steinberg, Frank Budinsky, Ed Merks, and Marcelo Paternostro. *EMF: eclipse modeling framework*. Pearson Education, 2008.

[200] Markus Voelter and Vaclav Pech. Language modularity with the MPS language workbench. In *Software Engineering (ICSE), 2012 34th International Conference on*, pages 1449–1450. IEEE, 2012.

[201] Alberto Rodrigues da Silva, Sinia Vlajic, Saa Lazarevic, Ilija Antovic, Vojislav Stanojevic, and Milo Milic. Preliminary experience using JetBrains MPS to implement a requirements specification language. In *Quality of Information and Communications Technology (QUATIC), 2014 9th International Conference on the*, pages 134–137. IEEE, 2014.

[202] Lennart C. L. Kats and Eelco Visser. *The spoofax language workbench: rules for declarative specification of languages and IDEs*, volume 45. ACM, 2010.

[203] Charles Simonyi, Magnus Christerson, and Shane Clifford. Intentional software. In *ACM SIGPLAN Notices*, volume 41, pages 451–464. ACM, 2006.

[204] Markus Voelter and Konstantin Solomatov. Language modularization and composition with projectional language workbenches illustrated with MPS. *Software Language Engineering, SLE*, 16:3, 2010.

[205] Sven Efftinge and Markus Völter. oAW xText: A framework for textual DSLs. In *Workshop on Modeling Symposium at Eclipse Summit*, volume 32, page 118, 2006.

[206] Moritz Eysholdt and Heiko Behrens. Xtext: implement your language faster than the quick and dirty way. In *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*, pages 307–309. ACM, 2010.

[207] Lorenzo Bettini. *Implementing Domain-Specific Languages with Xtext and Xtend.* Packt Publishing Ltd, 2013.

[208] Shane Sendall and Wojtek Kozaczynski. Model transformation: The heart and soul of model-driven software development. *IEEE software*, 20(5):42–45, 2003.

[209] James Howatt. A Project-based Approach to Programming Language Evaluation. *SIGPLAN Not.*, 30(7):37–40, July 1995. ISSN 0362-1340.

[210] Jarallah AlGhamdi and Joseph Urban. Comparing and Assessing Programming Languages: Basis for a Qualitative Methodology. In *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing: States of the Art and Practice*, SAC '93, pages 222–229, New York, NY, USA, 1993. ACM. ISBN 978-0-89791-567-0. doi: 10.1145/162754.162876.

[211] Melvin Klerer. *Design of very high-level computer languages: a user oriented approach.* McGraw-Hill, Inc., 1991.

[212] David A. Watt, William Findlay, and John Hughes. *Programming language concepts and paradigms*, volume 234. Prentice Hall Englewood Cliffs, 1990.

[213] Alan F. Blackwell, Luke Church, and T. R. G. Green. The abstract is' an enemy': Alternative perspectives to computational thinking. In *Proceedings PPIG*, volume 8, pages 34–43, 2008.

[214] Carlo Ghezzi and Mehdi Jazayeri. *Programming language concepts.* New York: Wiley, 1997.

[215] Shane Markstrum. Staking claims: a history of programming language design claims and evidence: a positional work in progress. In *Evaluation and Usability of Programming Languages and Tools*, page 7. ACM, 2010.

[216] Ankica Barišic, Vasco Amaral, Miguel Goulão, and Bruno Barroca. Evaluating the usability of domain-specific languages. In *Software Design and Development: Concepts, Methodologies, Tools, and Applications*, pages 2120–2141. IGI Global, 2014.

[217] Pedro Gabriel, Miguel Goulao, and Vasco Amaral. Do software languages engineers evaluate their languages? *arXiv preprint arXiv:1109.6794*, 2011.

[218] Mary Shaw, Guy T. Almes, Joseph M. Newcomer, Brian K. Reid, and Wm A. Wulf. A comparison of programming languages for software engineering. *Software: Practice and Experience*, 11(1):1–52, 1981.

[219] Richard B. Kieburtz, Laura McKinney, Jeffrey M. Bell, James Hook, Alex Kotov, Jeffrey Lewis, Dino P. Oliva, Tim Sheard, Ira Smith, and Lisa Walton. A software engineering experiment in software component generation. In *Proceedings of the 18th international conference on Software engineering*, pages 542–552. IEEE Computer Society, 1996.

[220] Niklaus Wirth. On the Design of Programming Languages. In *IFIP Congress*, volume 74, pages 386–393, 1974.

[221] Michael Pfeiffer and Josef Pichler. A comparison of tool support for textual domain-specific languages. In *Proceedings of the 8th OOPSLA Workshop on Domain-Specific Modeling*, pages 1–7, 2008.

[222] Christian Kastner, Thomas Thum, Gunter Saake, Janet Feigenspan, Thomas Leich, Fabian Wielgorz, and Sven Apel. FeatureIDE: A tool framework for feature-oriented software development. In *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*, pages 611–614. IEEE, 2009.

[223] Tomaž Kosar, Sudev Bohra, and Marjan Mernik. Domain-specific languages: A systematic mapping study. *Information and Software Technology*, 71:77–91, 2016.

[224] Dimitrios S. Kolovos, Richard F. Paige, Tim Kelly, and Fiona AC Polack. Requirements for domain-specific languages. In *Proc. of ECOOP Workshop on Domain-Specific Program Development (DSPD)*, volume 2006, 2006.

[225] Abdelilah Kahlaoui, Alain Abran, and E. Lefebvre. DSML success factors and their assessment criteria. *Metrics News*, 13(1):43–51, 2008.

[226] Juha Kärnä, Juha-Pekka Tolvanen, and Steven Kelly. Evaluating the use of domain-specific modeling in practice. In *Proceedings of the 9th OOPSLA workshop on Domain-Specific Modeling*, 2009.

[227] Parastoo Mohagheghi and Øystein Haugen. Evaluating Domain-Specific Modelling Solutions. In *Advances in Conceptual Modeling - Applications and Challenges*, number 6413 in Lecture Notes in Computer Science, pages 212–221. Springer Berlin Heidelberg, November 2010.

[228] Øystein Haugen and Parastoo Mohagheghi. A multi-dimensional framework for characterizing domain specific languages. In *Proceedings of the 7th OOPSLA Workshop on Domain-Specific Modeling (DSM'07), Computer Science and Information System Reports*, 2007.

[229] Yali Wu, Frank Hernandez, Francisco Ortega, Peter J. Clarke, and Robert France. Measuring the effort for creating and using domain-specific models. In *Proceedings of the 10th Workshop on Domain-Specific Modeling*, page 14. ACM, 2010.

[230] Gökhan Kahraman and Semih Bilgen. A framework for qualitative assessment of domain-specific languages. *Software & Systems Modeling*, 14(4):1505–1526, November 2013. ISSN 1619-1366, 1619-1374. doi: 10.1007/s10270-013-0387-8.

[231] Jeff Sauro and James R. Lewis. *Quantifying the user experience: Practical statistics for user research.* Elsevier, 2012.

[232] John Brooke. SUS: A Retrospective. *J. Usability Studies*, 8(2):29–40, February 2013. ISSN 1931-3357.

[233] William Albert and Thomas Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics.* Newnes, 2013.

[234] James R. Lewis. Measuring Perceived Usability: The CSUQ, SUS, and UMUX. *International Journal of Human–Computer Interaction*, pages 1–9, 2018.

[235] Timo Jokela, Netta Iivari, Juha Matero, and Minna Karukka. The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In *Proceedings of the Latin American conference on Human-computer interaction*, pages 53–60. ACM, 2003.

[236] Nigel Bevan, James Carter, and Susan Harker. ISO 9241-11 revised: What have we learnt about usability since 1998? In *International Conference on Human-Computer Interaction*, pages 143–151. Springer, 2015.

[237] Jum C. Nunnally. Psychometric theory. *Mc Graw-Hill Publ Co, New York*, 1978.

[238] James R. Lewis. Usability: Lessons Learned . . . and Yet to Be Learned. *International Journal of Human–Computer Interaction*, 30(9):663–684, September 2014. ISSN 1044-7318. doi: 10.1080/10447318.2014.930311.

[239] Ahlem Assila and Houcine Ezzedine. Standardized Usability Questionnaires: Features and Quality Focus. *Electronic Journal of Computer Science and Information Technology: eJCIST*, 6(1), 2016.

[240] Kasper Hornbæk and Effie Lai-Chong Law. Meta-analysis of correlations among usability measures. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 617–626. ACM, 2007.

[241] Jeff Sauro and Joseph S. Dumas. Comparison of Three One-question, Post-task Usability Questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1599–1608, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518946.

[242] John P. Chin, Virginia A. Diehl, and Kent L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–218. ACM, 1988.

[243] Jurek Kirakowski and Mary Corbett. SUMI: The software usability measurement inventory. *British journal of educational technology*, 24(3):210–212, 1993.

[244] Jurek Kirakowski. The software usability measurement inventory: background and usage. *Usability evaluation in industry*, pages 169–178, 1996.

[245] James R. Lewis. Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 36, pages 1259–1260. SAGE Publications Sage CA: Los Angeles, CA, 1992.

[246] James R. Lewis. Psychometric evaluation of the PSSUQ using data from five years of usability studies. *International Journal of Human-Computer Interaction*, 14(3-4):463–488, 2002.

[247] Ann Fruhling and Sang Lee. Assessing the reliability, validity and adaptability of PSSUQ. *AMCIS 2005 Proceedings*, page 378, 2005.

[248] John Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[249] James R. Lewis and Jeff Sauro. The factor structure of the system usability scale. In *Human centered design*, pages 94–103. Springer, 2009.

[250] Aaron Bangor, Philip Kortum, and James Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies*, 4(3): 114–123, May 2009. ISSN 1931-3357.

[251] Kraig Finstad. The usability metric for user experience. *Interacting with Computers*, 22(5):323–327, 2010.

[252] Mehmet Ilker Berkman and Dilek Karahoca. Re-assessing the usability metric for user experience (UMUX) scale. *Journal of Usability Studies*, 11(3):89–109, 2016.

[253] James R. Lewis, Brian S. Utesch, and Deborah E. Maher. UMUX-LITE: when there's no time for the SUS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2099–2102. ACM, 2013.

[254] Paul Cairns. A commentary on short questionnaires for assessing usability. *Interacting with Computers*, 25(4):312–316, 2013.

[255] John J. Bosley. Creating a short usability metric for user experience (UMUX) scale. *Interacting with Computers*, 25(4):317–319, 2013.

[256] National Statistical Office (NSO) and ICF Macro. Malawi demographic and health survey 2010. Technical report, The Malawi Government, 2010.

[257] WHO Regional Office for Africa. Second generation, who country cooperation strategy, 2008-2013, malawi. Technical report, The World Health Organisation, 2009.

[258] Jonathan A. Patz, Thaddeus K. Graczyk, Nina Geller, and Amy Y. Vittor. Effects of environmental change on emerging parasitic diseases. *International Journal for Parasitology*, 30(12):1395–1405, November 2000. ISSN 0020-7519. doi: 10.1016/S0020-7519(00)00141-7.

[259] Sari R. Kovats, Menno J. Bouma, Shakoor Hajat, Eve Worrall, and Andy Haines. El Niño and health. *The Lancet*, 362(9394):1481–1489, November 2003. ISSN 0140-6736. doi: 10.1016/S0140-6736(03)14695-8.

[260] Sonia Altizer, Andrew Dobson, Parviez Hosseini, Peter Hudson, Mercedes Pascual, and Pejman Rohani. Seasonality and the dynamics of infectious diseases. *Ecology Letters*, 9(4):467–484, April 2006. ISSN 1461-0248. doi: 10.1111/j.1461-0248.2005.00879.x.

[261] National Statistical Office (NSO) and ICF. Malawi demographic and health survey 2015-2016. Technical report, The Malawi Government, 2017.

[262] Henry Kankwamba, Mariam A. T. J. Mapila, and Karl Pauw. Determinants and spatiotemporal dimensions of crop diversification in Malawi. *Project Report produced under a co-financed research agreement between Irish Aid, USAID and IFPRI, Paper*, (3), 2012.

[263] Raymond G. Gordon and Barbara F. Grimes. *Ethnologue: Languages of the world*, volume 15. Sil international Dallas, TX, 2005.

[264] Gerald P. Douglas, Zach Landis-Lewis, and Harry Hochheiser. Simplicity and usability: Lessons from a touchscreen electronic medical record system in Malawi. *Interactions*, 18(6):50–53, 2011. ISSN 1072-5520. doi: 10.1145/2029976.2029990.

[265] Gordon Baxter and Ian Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23(1):4–17, 2011. ISSN 0953-5438, 1873-7951. doi: 10.1016/j.intcom.2010.07.003.

[266] Ian Sommerville. *Software Engineering.* Addison-Wesley, Harlow, England, 9 edition, 2010. ISBN 978-0-13-703515-1.

[267] Arshdeep Bahga and Vijay K. Madisetti. A cloud-based approach for interoperable electronic health records (ehrs). *IEEE Journal of Biomedical and Health Informatics*, 17(5):894–906, 2013.

[268] Shaker H. El-Sappagh and Samir El-Masri. A distributed clinical decision support system architecture. *Journal of King Saud University - Computer and Information Sciences*, 26(1):69–78, January 2014. ISSN 1319-1578. doi: 10.1016/j.jksuci.2013.03.005.

[269] Kenneth D. Mandl and Isaac S. Kohane. No small change for the health information economy. *New England Journal of Medicine*, 360(13):1278–1281, 2009. ISSN 0028-4793. doi: 10.1056/NEJMp0900411.

[270] Ignacio Valdes, David C. Kibbe, Greg Tolleson, Mark E. Kunik, and Laura A. Petersen. Barriers to proliferation of electronic medical records. *Informatics in Primary Care*, 12(1):3–9, 2004.

[271] Thomas R. Yackel. How the open-source development model can improve medical software. *Studies in Health Technology and Informatics*, 84:68–72, 2001. ISSN 0926-9630.

[272] Peter M. Yellowlees, Shayna L. Marks, Michael Hogarth, and Stuart Turner. Standards-based, open-source electronic health record systems: A desirable future for the U.S. health industry. *Telemedicine and e-Health*, 14(3):284–288, 2008. ISSN 1530-5627. doi: 10.1089/tmj.2007.0052.

[273] Ryszard S. Michalski. A theory and methodology of inductive learning. In Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning*, Symbolic Computation, pages 83–134. Springer Berlin Heidelberg, 1983. ISBN 978-3-662-12407-9 978-3-662-12405-5.

[274] Harvey F. Silver, R. Thomas Dewing, and Matthew J. Perini. *The Core Six: Essential Strategies for Achieving Excellence with the Common Core*. ASCD, 2012. ISBN 978-1-4166-1495-1.

[275] WHO Regional Office for Africa. Implementation of Option B+ for prevention of mother-to-child transmission of HIV: the Malawi experience. Technical report, The World Health Organisation, 2014.

[276] Colin Atkinson and Thomas Kühne. Model-driven development: a metamodeling foundation. *Software, IEEE*, 20(5):36–41, 2003.

[277] Steffen Staab, Tobias Walter, Gerd Gröner, and Fernando Silva Parreiras. Model driven engineering with ontology technologies. In *Reasoning web. semantic technologies for software engineering*, pages 62–98. Springer, 2010.

[278] Marco Brambilla, Jordi Cabot, and Manuel Wimmer. Model-driven software engineering in practice. *Synthesis Lectures on Software Engineering*, 1(1):1–182, 2012.

[279] Kevin Donnelly. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in health technology and informatics*, 121:279, 2006.

[280] Gergely Héja, György Surján, and Péter Varga. Ontological analysis of SNOMED CT. *BMC Medical Informatics and Decision Making*, 8(1):1–5, 2008. ISSN 1472-6947. doi: 10.1186/1472-6947-8-S1-S8.

[281] Heinrich Herre. General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In *Theory and applications of ontology: computer applications*, pages 297–345. Springer, 2010.

[282] Holger Knublauch, Ray W. Fergerson, Natalya F. Noy, and Mark A. Musen. The Protégé OWL plugin: An open development environment for semantic web applications. In *International Semantic Web Conference*, pages 229–243. Springer, 2004.

[283] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, October 2005. ISSN 0004-3702. doi: 10.1016/j.artint.2005.05.003.

[284] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau. OWL 2 web ontology language direct semantics. W3C recommendation, W3C, 27 October 2009 2009.

[285] Carl A. Gunter. *Semantics of programming languages: structures and techniques.* MIT press, 1992.

[286] Spencer P. Florence, Bruke Fetscher, Matthew Flatt, William H. Temps, Tina Kiguradze, Dennis P. West, Charlotte Niznik, Paul R. Yarnold, Robert Bruce Findler, and Steven M. Belknap. POP-PL: a patient-oriented prescription programming language. In *ACM SIGPLAN Notices*, volume 51, pages 131–140. ACM, 2015.

[287] Heiko Behrens, Michael Clay, Sven Efftinge, Moritz Eysholdt, Peter Friese, Jan Köhnlein, Knut Wannheden, and Sebastian Zarnekow. Xtext user guide. 2008. URL https://eclipse.org/Xtext/documentation/1_0_1/xtext.pdf.

[288] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Automata theory, languages, and computation. *International Edition*, 24, 2006.

[289] Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. In *Proceedings of the*

2016 CHI Conference on Human Factors in Computing Systems, CHI '16, pages 3227–3231, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7.

[290] Thomas Harkins, Christopher Drasbek, Juan Arroyo, and Michael McQuestion. The health benefits of social mobilization: experiences with community-based Integrated Management of Childhood Illness in Chao, Peru and San Luis, Honduras. *Promotion & Education*, 15(2):15–20, June 2008. ISSN 1757-9759, 1757-9767.

[291] Rosette J. Chakkalakal, Emily Cherlin, Jennifer Thompson, Tessa Lindfield, Ruth Lawson, and Elizabeth H. Bradley. Implementing clinical guidelines in low-income settings: A review of literature. *Global Public Health*, 8(7):784–795, August 2013. ISSN 1744-1692. doi: 10.1080/17441692.2013.815794.

[292] Terence Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Programmers, 2007.

[293] Viswanath Venkatesh, Susan A. Brown, and Hillol Bala. Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS quarterly*, 37(1), 2013.

[294] Thomas S. Tullis and Jacqueline N. Stetson. A comparison of questionnaires for assessing website usability. In *Usability professional association conference*, pages 1–12, 2004.

[295] Kraig Finstad. The system usability scale and non-native english speakers. *Journal of usability studies*, 1(4):185–188, 2006.

[296] Terrence W. Pratt, Marvin V. Zelkowitz, and Tadepalli V. Gopal. *Programming languages: design and implementation*. 1984.

[297] Richard F. Paige, Jonathan S. Ostroff, and Phillip J. Brooke. Principles for modeling language design. *Information and Software Technology*, 42(10):665–675, 2000.

[298] Alan F. Blackwell and Cecily Morrison. A logical mind, not a programming mind: Psychology of a professional end-user. In *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group (PPIG 2010)*, pages 19–22, 2010.

[299] Matthias Gietzelt, Ursula Goltz, Daniel Grunwald, Malte Lochau, Michael Marschollek, Bianying Song, and Klaus-Hendrik Wolf. Arden2bytecode: a one-pass Arden Syntax compiler for service-oriented decision support systems based on the OSGi platform. *Computer methods and programs in biomedicine*, 106(2): 114–125, 2012.

[300] Arnoud A. F. van der Maas, Arthur H. M. ter Hofstede, and A. Johannes ten Hoopen. Requirements for Medical Modeling Languages. *Journal of the American Medical Informatics Association*, 8(2):146–162, March 2001. ISSN 1067-5027, 1527-974X. doi: 10.1136/jamia.2001.0080146.

[301] Bayo Lawal. Repeated Measures Design. In *Applied Statistical Methods in Agriculture, Health and Life Sciences*, pages 697–718. Springer International Publishing, 2014. ISBN 978-3-319-05554-1 978-3-319-05555-8.

[302] Scott I. MacKenzie. *Human-computer interaction: An empirical research perspective.* Newnes, 2012.

[303] YoungSeok Yoon and Brad A. Myers. Capturing and analyzing low-level events from the code editor. In *Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools*, pages 25–30. ACM, 2011.

[304] Jeff Sauro. The challenges and opportunities of measuring the user experience. *Journal of Usability Studies*, 12(1):1–7, 2016.

[305] Erik Frøkjær, Morten Hertzum, and Kasper Hornbæk. Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 345–352, New York, NY, USA, 2000. ACM. ISBN 978-1-58113-216-8. doi: 10.1145/332040.332455.

[306] Nigel Bevan. Measuring usability as quality of use. *Software Quality Journal*, 4 (2):115–130, June 1995. ISSN 0963-9314, 1573-1367. doi: 10.1007/BF00402715.

[307] E. Morozoff. Using a Line of Code Metric to Understand Software Rework. *IEEE Software*, 27(1):72–77, January 2010. ISSN 0740-7459. doi: 10.1109/MS.2009.160.

[308] Graham R. Gibbs. *Analysing qualitative data.* Sage, 2008.

[309] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[310] Odd Ivar Lindland, Guttorm Sindre, and Arne Solvberg. Understanding quality in conceptual modeling. *IEEE software*, 11(2):42–49, 1994.

[311] Martin Butzlaff, Horst Vollmar, Bettina Floer, Nik Koneczny, Jana Isfort, and Silke Lange. Learning with computerized guidelines in general practice? A randomized controlled trial. *Family Practice*, 21(2):183–188, 2004.

[312] Monika Moser and Silvia Miksch. Improving Clinical Guideline Implementation Through Prototypical Design Patterns. In *Artificial Intelligence in Medicine*, pages 126–130. Springer, Berlin, Heidelberg, July 2005. doi: 10.1007/11527770_17.

[313] Rong Chen and Iago Corbal. Guideline Definition Language (GDL), 2013. URL http://www.openehr.org/news_events/releases/releases.php?id=79.