CONCEPTUAL MODELING PRACTICE IN AGILE DEVELOPMENT METHODOLOGIES

AN EXPLORATORY STUDY INTO THE UNDERLYING ASSUMPTIONS OF CONCEPTUAL MODELING PRACTICE

By

Naghmeh Sharikzadeh

Thesis
submitted to the
Faculty of Information Technology
in fulfillment of the requirements for the degree of
Master of Information Technology (by Research)
at
Monash University

Melbourne 2015

STATEMENT OF ORIGINAL AUTHORSHIP

I hereby declare that, to the best of my knowledge and belief, this thesis entitled *STATUS OF CONCEPTUAL MODELING PRACTICE IN AGILE METHODOLOGY: AN EXPLORATORY STUDY INTO THE UNDERLYING ASSUMPTIONS OF MODELING PRACTICE* is my own original work. The work presented in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institute. Due acknowledgement to each significant contribution to, and quotation in this thesis from the work, or works of other people has been made through the proper use of citations and references.

Melbourne, 1 May 2015.

[redacted]

(Naghmeh Sharikzadeh)

Supervisory Panel

Principal Supervisor

Professor Ron Weber
Monash University
University of Queensland

Associate Supervisor

Dr. Caddie Gao
Faculty of Information Technology
Monash University

ABSTRACT

Conceptual modeling is the practice of formally describing a real-world domain to enable better understanding and communication among stakeholders. It provides a basis for design during the development of information systems.

In a sequential approach to system development projects (e.g., via traditional waterfall methodologies), *a priori* specification of system requirements via complete, clear conceptual modeling scripts is deemed essential to the success of a project. In the current practice of system development, however, non-sequential methodologies such as agile methodologies are gaining increasing prominence. Conceptual modeling supposedly does not have a primary role to play in these methodologies. Nonetheless, use of these methodologies seems not to have increased failure rates in system development projects, even though they are deemed to downplay the role of conceptual modeling.

My research uses an ontological perspective on conceptual modeling to explore the practice of system development when agile methodologies are used. I conducted an interpretive field study involving semi-structured interviews with eight highly experienced practitioners to explore the context and methods of conceptual modeling in agile settings. I provide an explanation of the anomalies that exist in the perceived role of conceptual modeling when using agile methodologies, compared to waterfall methodologies.

Based on the findings of my study, I have concluded that contrary to much current rhetoric, the practice of conceptual modeling is not becoming obsolete in agile methodologies. Rather, its importance is growing. This outcome is occurring with

agile methodologies for two reasons. First, information is increasingly recognised as a concrete asset that brings value to organisations. Therefore, procedures such as conceptual modeling that enable information about domain semantics to be extracted more easily are gaining significance. Second, advances in technological infrastructures have enabled practitioners to focus more on a domain's semantics as many system implementation details are becoming standardised.

In spite of growing significance of conceptual modeling, the findings of my study indicate that fundamental differences exist between conceptual modeling practice in agile methodologies and traditional sequential methodologies. The differences are twofold. First, agile methodologies differ from sequential methodologies in terms of the level of *granularity* of the conceptual models used to represent domain semantics. In agile methodologies, detailed, *a priori* specifications of domain semantics are not needed. Instead, domains are often represented only in terms of their main subject matter. In this regard, conceptual models in agile methodologies are deemed to be coarse-grained representations of domains (compared to their sequential counterparts that attempt to provide complete, finely grained representations of domains).

Second, the results of my study show that the practice of conceptual modeling is influenced by the context and overall objective of the information system for which the conceptual models are developed. While formalisation of conceptual modeling in traditional methodologies seems to be unaffected by different types of information systems that exist, a theme of System Taxonomy emerged through the analysis of data in my study. This theme indicates that domain uncertainty and volatility, as well as the overall objectives of different information systems, influence the practice of conceptual modeling in agile system development.

My study makes contributions to the body of knowledge through development of new concepts and provision of rich insights about real-life practices of conceptual modeling. It also expands the boundaries of current theories about conceptual modeling (initially developed in sequential settings) by showing their relevance, at least in part, in non-sequential settings.

ACKNOWLEDGEMENTS

I wish to extend my sincere gratitude to my supervision panel, Professor Ron Weber and Dr. Caddie Gao. It was an absolute pleasure to work with them. I am particularly privileged for the opportunity and honor of having worked with Ron and having received his diligent supervision and outstanding mentorship. His significant guidance, support, criticism, wisdom, and motivation have been invaluable to me. My aspiration for fine scholarship in the field of information systems arises from working with him, and it all means so much to me.

I also wish to thank John Giles for sharing his thorough and critical approach to development of information systems with me. I was fortunate to learn about his perspective on system development during a brief time. However, his ideas had an enduring impact on my understanding of this practice, and I thank him for this outcome.

Writing this thesis opened up incredible horizons to me. Through the experience, I learned about myself more than any other experiences I have ever had in life. Among all, one was especially revelatory and left some mark of humor on the path! I realised that in particular instances, how easily and unexpectedly I can become tearful! Thinking of all those teary moments—some out of joy and some not so—I wish to particularly thank my beloved husband, Hamid, for all his support along the way. I am truly thankful to him, not just for his emotional support, but also for being a wonderful source of intellectual stimulation and guidance to me. Without his support, this journey would have been lonely, at times dreadful, and impossible.

This note will be incomplete if I do not name my beautiful little son, Mehrbod. He was my closest company throughout this journey. He persistently inspired me with his sense of awe toward the world. I thank him wholeheartedly for the cherished company of his precious life, and for reminding me that no challenge compares to the challenge of raising him. I thank him for constantly empowering me by giving me the right perspective.

Last but not least, I wish to thank my mother and father whom I have been living far from for a few years now. Their ethics and the love they have given me have made me who I am, and will shape who I become. I thank them for teaching me resilience and for giving me a great sense and admiration for high morality. These two assets have made my life meaningful and abundant.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This chapter presents the motivation for my study and elucidates the research questions I attempt to answer. I explain the rationale for choosing a qualitative interpretive approach to answer the proposed research questions. Finally, I explain the organisation of this thesis.

1.1 Background

Development of information systems is a core practice in the field of Information Systems. As a result, understanding the context, conduct, and implications of system development methodologies has had a major influence on theoretical and empirical research undertaken in the field.

A key element in the development of information systems that has attracted researchers' attention is the practice of *conceptual modeling*. Traditionally, conceptual modeling has been defined as the practice of formally describing a real-world domain to enable better understanding and communication among stakeholders.

For two reasons, conceptual modeling has been recognised among the most important tasks undertaken during development of information systems. First, conceptual models enable integration of domain experts' knowledge into system development practices. Second, conceptual models fundamentally unburden stakeholders from having to consider tedious implementation details when they initially try to understand a domain (Roussopoulos & Karagiannis, 2009).

Conceptual modeling is predominantly understood as a distinct, visible practice that takes place during the early stages of system development. Furthermore, the completeness and accuracy of the conceptual modeling scripts (models) that are produced are thought to determine the quality of the information system that is designed based on these scripts (Burton-Jones, Wand, & Weber, 2009; Krogstie, Lindland, & Sindre, 1995; Lindland, Sindre, & Solvberg, 1994; Nelson, Poels, Genero, & Piattini, 2012; Roussopoulos & Karagiannis, 2009; Wand, Monarchi, Parsons, & Woo, 1995; Wand & Weber, 2002; Weber, 1997).

These perceptions of conceptual modeling scripts and their role in the success of information systems, however, largely reflect the practice of traditional system development methodologies, such as *waterfall* (Royce, 1970). In waterfall methodologies, system development follows a sequential approach with an emphasis on documenting every stage. Attaining an ever-comprehensive initial design based on high-quality conceptual modeling scripts drives the practice of modeling in waterfall methodologies. The objective is to prevent or minimise future errors in the system (Phatak, 2012).

The assumptions that underpin sequential system development methodologies such as waterfall are now under scrutiny (Lukyanenko & Parsons, 2013a). First, some alternative system development methodologies do not require a preliminary complete and clear specification of domains. In *agile methodologies*, for instance, minimal emphasis is placed on documentation. Moreover, formal conceptual modeling scripts are no longer used to initiate the design (Ambler, 2005; Conboy, 2009; Lindstrom & Jeffries, 2004). Additionally, some research shows that an increasing number of information system projects are adopting alternative methodologies (Benediktsson, Dalcher, & Thorbergsson, 2006). Nonetheless, lack of conformity to traditional conceptual modeling practices appears to have no correlation with the success of information system projects (Pederson, 2013).

Second, the emergence of a new class of systems—namely, *open systems*—poses serious challenges to the principles and theoretical frameworks that currently inform conceptual modeling. These challenges undermine the possibility and desirability of having a preliminary specification of requirements (as in traditional methodologies). For instance, new information systems developments, such as semantic search engines, social networking, and crowdsourcing, are no longer based on *abstraction* of information (as manifested in traditional methodologies via class diagrams and entity-relationship models) (Parsons & Wand, 2000). Moreover, they do not seek to reconcile stakeholders' divergent views of a domain as expressed in sets of formal conceptual modeling scripts (system specifications). Instead, open systems are based on distributed, heterogeneous user information and draw upon unique as opposed to general, abstracted information (Lukyanenko & Parsons, 2013a; Roussopoulos & Karagiannis, 2009).

As a consequence, lack of correlation between information system success and the system development methodologies used, as well as the emergence of new classes of information systems, challenge the conventional understanding of conceptual models as a basis for system design. These challenges motivate the need to obtain a deeper understanding of the role of conceptual modeling in practice and its theoretical foundations. Obtaining such an understanding is important because it could provide a theoretical explanation as to why alternative methodologies are replacing traditional methodologies in some system development projects. Moreover, it could lead to the identification of those circumstances in which alternative methodologies are replacing traditional ones. In short, a deeper understanding of the circumstances leading to use of alternative modeling practices potentially provides insights about the distinctive features of different types of information systems. These insights can have important implications for the theory and practice of information systems development.

In this light, this study seeks to obtain a deeper understanding of conceptual modeling practice and its theoretical foundations. For two reasons, I have chosen to investigate the practice of conceptual modeling in *agile* methodologies (as an alternative setting to waterfall methodologies.) First, studying an alternative setting helps to give insight about the different circumstances that impact conceptual modeling in practice. Second, I wish to address a less studied area in conceptual modeling research—while the implications of abstraction and generalisation methods in conventional conceptual modeling have gained considerable attention (Lukyanenko & Parsons, 2011a, 2011b, 2013b; Parsons & Wand, 2000), few studies focus on exploring the practice of conceptual modeling in alternative methodologies, particularly from a theoretical perspective.

1.2 Problem Statement and Purpose of Study

Having high-quality conceptual models is of great importance in traditional system development methodologies. Indeed, the nature of and the methods for achieving high-quality conceptual models have been the subject of much empirical and theoretical research (Burton-Jones et al., 2009; Krogstie et al., 1995; Lindland et al., 1994; Nelson et al., 2012; Roussopoulos & Karagiannis, 2009; Wand et al., 1995; Wand & Weber, 2002; Weber, 1997).

On the other hand, the literature on agile methodologies indicates that agile does not emphasise formal documentation. Instead, practitioners in agile methodologies strive to deliver working software rather than formal conceptual modeling scripts as a basis for design (Ambler, 2005; Highsmith & Cockburn, 2001; Turk, Robert, & Rumpe, 2005). The literature is not clear, however, about whether the practice of conceptual modeling in agile methodologies has been abandoned completely or has been transformed in some ways or has simply become *invisible*. Clarifying these points constitutes the essence of my first research question:

Research Question 1:

To what extent is conceptual modeling done when agile methodologies are used to develop information systems?

I endeavour to answer this question by conducting an exploratory study on practitioners' views about conceptual modeling in agile methodologies. By way of conducting this exploratory study, I am able to address a more general question:

Research Question 2:

What are practitioners' views about the need for and importance of conceptual modeling when they use agile methodologies to develop information systems?

1.3 Research Approach and Methodology

As discussed above, this study seeks to obtain a deeper understanding of the role of conceptual modeling in the context of agile methodologies. It is motivated by the challenges that newly emerging open systems and alternative system development methodologies propose to the conventional understanding of conceptual modeling. Against this background, I argue that a *qualitative interpretive research methodology* is best suited to address the two research questions proposed.

Kaplan and Maxwell (2005) argue that *qualitative* methods provide a strong research approach in situations where researchers are to “examine the dynamics of a process” (p. 31) and understand “the meaning and the context of the phenomena studied, and the particular events and processes that make up these phenomena over time, in real-life, natural settings” (p. 31). They explain that “qualitative studies may begin with specific concerns or even suppositions about what is going on; seeing the unexpected, disconfirming one’s assumptions, and discovering new ways of making sense of what is going on” (Kaplan & Maxwell, 2005, p. 36).

In this light, my study explores the context of conceptual modeling practice *qualitatively*. By interviewing practitioners who have experience in agile projects, I endeavour to answer my two research questions based on the *real-life*, *natural-setting* experiences of practitioners who are engaged in system development projects. The two research questions do not require true or false answers. Instead, they lead to *narrower*, more *focused* open-ended questions that I can ask of practitioners. In this way, they enable me to understand *the dynamics of the process* of conceptual modeling from practitioners’ perspectives (Kaplan & Maxwell, 2005).

As the background of my study is motivated by specific challenges to the traditional assumptions that underpin conceptual modeling, I choose a qualitative research approach to gain an understanding about practitioners' perspectives on conceptual modeling. I seek to explain the nature of conceptual modeling and the contextual elements impacting it. Moreover, based on Klein and Myers (1999), I argue that my work is best done as an *interpretive field study.*

In this regard, once lack of consensus about theoretical assumptions arises, Burton-Jones (2012) argues that an *interpretive study* of the core ideas about the phenomena of interest is important. As discussed earlier, newly emerging open systems and alternative system development methodologies have created new challenges for the conventional understanding of conceptual modeling. These challenges have led to a lack of consensus that now exists about the role of conceptual modeling in alternative information systems and methodologies. In this regard, agile proponents argue that complete *a priori* specification of domain semantics is no longer desirable. Furthermore, the assumption of a correlation between success in information systems development and *a priori* domain specification has also been questioned. Based on Burton-Jones's (2012) rationale, I argue an interpretive study of fundamental assumptions about conceptual modeling could potentially lead to the generation of new hypotheses and greater insights about existing theories of conceptual modeling.

Moreover, Whetten (1989) underscores the importance of investigating applications of theories in "qualitatively different conditions" (p. 493). He explains how application of theories in alternative new settings establishes a *theoretical feedback loop,* potentially leading to new theoretical contributions. Specifically, he maintains how qualitatively investigating alternative settings leads to potential theoretical contributions by providing *explanations* of *why* anomalies exist. He states, "applying an old model to a new setting and showing that it works as expected is not instructive by itself. This conclusion has theoretical merit only if something about the new setting suggests the theory shouldn't work under those conditions" (p. 493).

In this light, I am undertaking a *qualitative interpretive* study of the real-life practice of conceptual modeling in agile methodologies. In my study, agile methodologies are the *alternative new setting*. The context of agile methodologies is qualitatively different from traditional methodologies. For instance, they do not emphasise conceptual modeling as a design activity based on heavy documentation. Nonetheless, our theoretical assumptions about conceptual modeling still seem to be influenced by the context of traditional methodologies. As a consequence, I expect that applying traditional theoretical assumptions about conceptual modeling in the new setting of agile methodologies will potentially contribute to theory by extending the boundaries of current theories about conceptual modeling.

Based on my study, I expect to explain current challenges and anomalies regarding conceptual modeling. Furthermore, through qualitatively exploring agile settings, I expect to contribute to practice by identifying contextual elements that impact the practice of conceptual modeling. Identifying these contextual elements can potentially inform future guidelines about suitable conceptual modeling practices and their fit in relation to different types of system development projects.

1.4 Organisation of Thesis

I present my research in five chapters. Following this introductory chapter, I provide a research overview in Chapter 2. I first examine literature about conceptual modeling and its role in developing information systems. I then discuss an ontological framework that has been used to understand conceptual modeling. I delineate the theoretical lens used in this study and discern gaps in knowledge based on this framework. Also, by discussing competing explanations of the role of conceptual modeling in improving domain understanding, I provide a problematised perspective of current understanding of conceptual modeling. Last, by juxtaposing the gaps and the discerned problematisation with implications for emerging information systems and alternative system development methodologies, I underscore the importance of qualitatively exploring the assumptions that underlie conceptual modeling practice. To set the background of my study, I close this chapter by a literature review on the waterfall and agile system development methodologies.

Chapter 3 describes the research methodology I used in my study. I first explain the rationale for my research as an interpretive field study. I then provide a detailed report on how I conducted semi-structured interviews to collect data, including sampling strategies and information about research participants. Last, I discuss hermeneutics and thematic analysis as the data analysis methods I used in my research.

In Chapter 4, I demonstrate how I used thematic analysis of data to identify emerging patterns in data, as themes. After defining and naming themes based on thematic analysis, I present and discuss these themes. As I attempt to provide a cohesive understanding of the complexity of conceptual modeling practice, I rely on principles of hermeneutics. Chapter 4 closes by recapitulating the findings of my study in describing contextual elements and methods of conceptual modeling practice in agile settings.

Chapter 5 concludes my thesis. I first summarise the research process and its outcomes. I then discuss the contributions of my research and its implications for theory and practice. Next, I elaborate on the strengths and limitations of my research in relation to methodology, theory, and data collection methods. Finally, I give some recommendations for future research.

2 RESEARCH OVERVIEW

This chapter presents the background to my research. First, I review literature that focuses on the significance of conceptual modeling in developing information systems. Second, I discuss the Ontological Perspective on Conceptual Modeling as the theoretical framework I use in this research. Third, I provide a review of the literature on waterfall and agile methodologies and discuss the assumptions that inform practice in each of these methodologies. Finally, I discuss the framework I used to undertake the interpretive study aimed at allowing me to obtain deeper insights about the role of conceptual modeling in the practice of information system development.

2.1 Research Background

Conceptual modeling has been discussed from various perspectives in Computer Science, Knowledge Representation, Artificial Intelligence, Databases and Requirements Analysis in Software Engineering, and Information Systems (Mylopoulos, 1998). The body of knowledge in conceptual modeling initially developed through designing and implementing a proliferation of modeling techniques rather than through theory (Wand & Weber, 2002). As the field was maturing, later theoretical attempts in Information Systems to address conceptual modeling phenomena were founded on different perspectives (Agarwal, Sinha, &

Tanniru, 1996; Khatri, Vessey, Ramesh, Clay, & Park, 2006; Lindland et al., 1994; Lyytinen, 1987; Siau & Rossi, 2011; Weber, 1997). Among these perspectives, a theoretical framework developed by Wand and Weber (1988, 1990, 1993, 1995) based on Bunge's Ontology (Bunge, 1977) is recognised as one of the more influential theories in conceptual modeling (Moody, 2009). This framework underpins a long-running program in Information Systems research (Burton-Jones, 2012). This thesis, henceforth, adopts Wand and Weber's ontological framework as its theoretical lens. I provide the reasons in the subsections below.

2.1.1 Conceptual Modeling in Information Systems Development

Conceptual modeling is defined as formally describing some aspects of the physical and social world—the *real world.* The practice of conceptual modeling aims at better understanding the real-world domain and improved communication among stakeholders as a result of better domain understanding (Mylopoulos, 1992).

Weber (2003a) describes conceptual modeling as building a representation of someone's perceptions of selected semantics about the real world. Wand et al. (1995) define conceptual models—as outcomes of the practice of conceptual modeling— to be abstract descriptions of the knowledge related to a real-world domain. Thalheim (2012) recognises conceptual modeling as a widely accepted practice in computer science. He stresses that conceptual modeling requires apprenticeship, technology, and design science. He contends that model development for large systems requires well-organised modeling processes, which are "evolution-prone and revision-prone" (Thalheim, 2012, p. 14). Roussopoulos and Karagiannis (2009) argue that conceptual modeling schema should evolve continuously and concurrently with other system-development activities. They maintain that users should learn and execute this evolution through a process that is not yet well understood. Burton-Jones and Meso (2006) identify a key activity in system development to be conceptualising the domain and representing it via a conceptual model.

The development of information systems conventionally involves a requirements analysis stage. By definition, Wand and Weber (2002) argue that the requirements analysis stage does not and should not canvass implementation concerns. Instead, conceptual modeling should be concerned with understanding the real-world domain. From this perspective, conceptual modeling is the process of eliciting and articulating domain requirements.

In traditional system development methodologies, analysis of domain requirements usually occurs during the early stages of system development. In traditional methodologies, *conceptual modeling scripts*, which are usually shown graphically or textually, represent the set of elicited and articulated requirements. These scripts are generated via *conceptual modeling grammars*. Conceptual modeling grammars comprise sets of grammatical constructs and rules, often in the form of a *modeling language*. In a modeling language, *conceptual modeling methods* describe how the grammatical constructs and rules should be used to build a model of a real-world domain. The practice of conceptual modeling and the use of conceptual modeling scripts take place in a *context.* This context is characterised principally by differences in stakeholders' cognitive abilities, differences in the targeted tasks to be accomplished by the system, and a broader setting of organisational change.

Wand and Weber (2002) enumerated four objectives for the process of conceptual modeling—namely, gaining a better understanding of the domain, providing a basis for design, documentation of original requirements, and improving stakeholders' communications. Among these objectives, better domain understanding is particularly important to researchers because it is understood to determine the success of the final information system. In particular, researchers argue that improved domain understanding reduces the likelihood of costly errors occurring in later stages of system development (Burton-Jones et al., 2009; Khatri et al., 2006; Offen, 2002). For instance, Moody (2005) contends that more than half of errors in system development projects relate to defects in system requirements analysis.

System requirements are elicited and articulated during the practice of conceptual modeling. If requirements errors occur, often they are too expensive to correct at the final stages of development because costs accumulate toward the end of projects. Therefore, if requirements errors are identified in the early stages of conceptual modeling during system development, they are more likely to be rectified. Moreover, errors that are not detected and rectified in the early stages of system development tend to carry over to other stages and affect the quality of the ultimate system. These errors can therefore lead into system failure. As a result, providing high-quality conceptual models is regarded as critical to successful system development projects.

2.1.2 Ontological Perspective on Conceptual Modeling – Representational Fidelity

Against this background, Wand and Weber's ontological framework for conceptual modeling focuses on the identification of deficiencies in conceptual modeling scripts and grammars (Wand & Weber, 1990, 1993, 1995). Weber (2003a) explains how adopting an ontological perspective on conceptual modeling provides a lens to better understand real-world domains. He argues that ontology formally explores and describes the structure of real-world objects—whether real-world objects are believed to be independent physical realities or some constructions of humans' perception. Therefore, by using ontology to impose order and meaning, understanding real-world domains will improve.

In the ontological framework of conceptual modeling, better domain understanding is postulated to occur when *representational fidelity* exists. Weber (2003b) argues that information systems are representations of some real-world domain. They are developed because observing the represented real-world domain in the form of an information system is either a better way or the only way to gain insight about the real-world domain.

For an information system to provide better understanding of its represented real-world domain, however, it must be a *faithful* representation of the domain. In other words, the quality of information systems in this framework depends on how faithfully they represent the underlying domain semantics. Faithful representations that are the result of better domain understanding in turn lead to improved understanding of the domain.

In this framework, the Theory of Ontological Expressiveness (TOE) (Burton-Jones & Weber, 2014), and the Theory of Logical Completeness (TLC) (Clarke, Burton-Jones, & Weber, 2013) formalise how representational fidelity in conceptual modeling scripts and grammars can be obtained. Both theories have implications for conceptual modeling methods because they are based on certain assumptions about the context of conceptual modeling practice. In the following subsections, I discuss some of these implications with respect to conceptual modeling methods, context, and the interplay between the method and context.

2.1.3 Conceptual Modeling Methods, Context and their Interplay – The Gap

Based on Wand and Weber's (2002) proposed research framework (shown in Figure 2-1), formalisation of conceptual modeling methods and context, using an ontological perspective, is under researched. While empirical evaluations of conceptual modeling grammars, scripts, and their interplay have been undertaken to some extent (Moody, 2009), the interplay of conceptual modeling context and methods has been relatively neglected. In this regard, Weber (2003a) states that "often we lack a formal understanding of why some practices seem effective and why others seem ineffective" (p.14). He asserts that adoption of different conceptual modeling methods in various contexts is not supported by any scientifically based explanations.

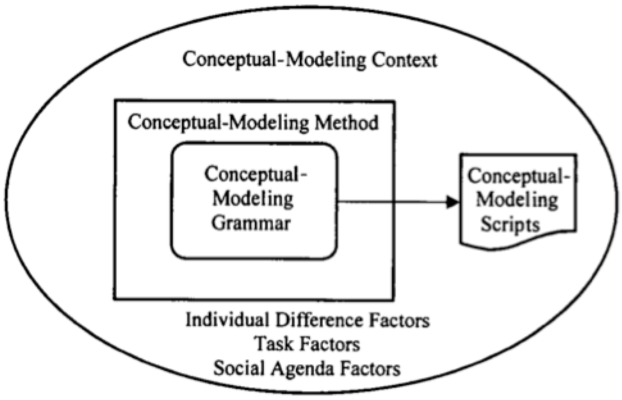

Figure 2-1 Conceptual modeling research framework adapted from Wand and Weber (2002, p. 364)

Methods of information requirements elicitation provide structure for formulating representations of the task environment (Davis, 1982). Representations of the task environment by definition underpin conceptual modeling scripts (Weber, 1997). Therefore, as Davis (1982) asserts, conceptual modeling scripts are impacted by methods that are chosen for requirements elicitation. He also emphasises the use of "nondata issues such as context, organisational policy, and roles" (p.11) by analysts in formulating the problem space, particularly in the presence of uncertainty in complex domains. Identifying these sources of *nondata issues* within organisations seems to correspond to Wand and Weber's (2002) social agendas, individual differences, and task factors—elements that shape the *context* of conceptual modeling. Therefore, as Davis (1982) underlines the importance of *nondata issues* (contextual elements), particularly in highly uncertain and complex domains, it is important to understand the impact of these contextual elements in highly uncertain and complex domains. I seek this understanding using the ontological framework of Wand and Weber (2002).

In extending Davis's recognition of the impacts of contextual element on methods of system development, however, a difference must be noted. Davis (1982) distinguishes between elicitation and evaluation methods in system development projects. He primarily focuses on elicitation methods and proposes that establishment of requirements is conducted at two levels. One is the organisational level to provide an overall depiction of system boundaries and interfaces. The other is the application level to provide a detailed specification of the objectives and assumptions of the subsystems.

Contrary to Davis, the proposed ontological framework of conceptual modeling is relevant to both elicitation and evaluation methods. Based on the proposed ontological framework, improvement of domain understanding is achieved in two ways. First, the ontological framework provides insights about the structures needed to organise and link real-world phenomena so that better ways to elicit domain requirements are identified. Second, it provides criteria for evaluating conceptual modeling scripts and grammars (Burton-Jones et al., 2009; Weber, 1997, 2003a).

Therefore, the ontological framework of conceptual modeling is relevant to both elicitation and evaluation of system requirements. Nevertheless, in spite of some rich empirical research that has been undertaken on the evaluation of conceptual models using TOE, the theoretical and practical implications of the ontological perspective on system requirements elicitation are not fully understood.

Weber (1997, 2003a) contends that application of the ontological perspective to conceptual modeling allows researchers to tease out questions that need to be answered about a domain. Nonetheless, elicitation of requirements based on the ontological perspective is under researched. Therefore, as a gap in the knowledge, it is not clear how elements of context impact methods of conceptual modeling elicitation and evaluation, especially in highly uncertain and complex domains.

2.1.4 Uncertainty and Complexity in Domains – A Competing Explanation on the Role of Conceptual Modeling in Domain Understanding

As the interplay of conceptual modeling context and methods in highly uncertain and complex domains is established as a gap in knowledge, I provide an overview of the extant literature.

Davis (1982) recognises *uncertainty* as a critical element in determining suitable strategies for requirements elicitation. Uncertainty, in his view, is illustrative of the complexity of a project's context and arises as a result of three factors: instability or non-existence of a set of requirements, cognitive constraints in specification of the requirements, and cognitive constraints in evaluating correctness and completeness of the elicited requirements. Based on the notion of *bounded rationality*, he argues that humans have limited capacity for rational thinking. For better domain understanding in complex contexts, therefore, humans construct *simplified* models of domains.

However, Davis (1982) argues that *simplification* of domains that arises as a result of humans' bounded rationality further constrains domain understanding in complex contexts.

This argument provides a *competing explanation* (Sandberg & Alvesson, 2011) for the perceived role of conceptual modeling, at least in highly complex and uncertain contexts. As discussed earlier, in the ontological framework of conceptual modeling, conceptual models are representations of domain semantics. Moreover, faithful representations of domain semantics based on this framework improve domain understanding. In other words, the ontological framework of conceptual modeling proposes that a conceptual model's representational fidelity is positively related to domain understanding.

In complex domains, however, Davis (1982) proposes that better domain understanding is constrained by a represented model's simplification. This understanding of the role of conceptual models competes with the explanation provided by the ontological perspective about the role of conceptual modeling. On the one hand, the ontological perspective proposes that domain understanding is improved as a result of conceptual modeling practice, provided that representational fidelity is achieved. On the other hand, based on humans' bounded rationality, Davis (1982) proposes that domain understanding is constrained by the representational simplification of conceptual models. Figure 2-2 sketches a diagram of this competing explanation of the role of conceptual modeling.

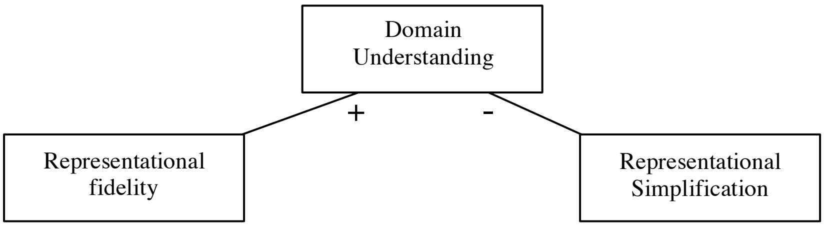

Figure 2-2 Competing explanation for role of conceptual modeling in complex domain understanding

While the interplay of the context and method in conceptual modeling forms a gap in knowledge, the proposed competing explanation problematises the extant literature in this area. As Sandberg and Alvesson (2011) explain, the existence of a competing explanation in the extant literature highlights anomalies that exist in our understanding of the role of conceptual modeling in information systems. In the following subsections, I further elaborate on the issue of *representational simplification* based on the extant literature. I then discuss a major anomaly that exists in our understanding of the role of conceptual modeling in emerging new information systems and alternative system development methodologies.

2.1.5 Ontological Perspective of Conceptual Modeling – Representational Simplification

In the practice of conceptual modeling, humans' bounded rationality suggests that representational simplification could hinder better domain understanding in highly complex domains. This exposition seemingly explains the controversy over the perceived role of conceptual modeling in improving domain understanding. In the ontological framework of conceptual modeling, improvements in domain understanding are formalised through the notion of representational fidelity. The logical completeness and ontological expressiveness of conceptual modeling grammars and scripts underpin faithful representation of domains.

However, the notion of representational simplification is not discussed explicitly in the ontological framework of conceptual modeling. Nonetheless, some modes of domain simplification have been investigated under this framework. For instance, Parsons and Wand (2000) extensively discuss theoretical and practical issues that are associated with *abstraction* and *classification,* which are dominant methods in modeling grammars. Although classifying objects is a pervasive cognitive activity, they explain that the underlying assumption of inherent classification raises serious issues for information modeling—particularly in the presence of multiple domain viewpoints, evolving domains as a result of changing stakeholder views, or information integration from different resources. They demonstrate how classifying objects into simplified representations can potentially inhibit better domain understanding.

Gemino and Wand (2005) also provided some insights regarding the notion of representational simplification. They show empirically how use of mandatory properties and subtypes in conceptual models improves domain understanding. This study has particular importance for the ontological framework of conceptual modeling because it shows how theories of cognition can be used to support an argument that representational clarity leads to better domain understanding. Moreover, this study leads to a counter-intuitive result. It shows how an increase in the overall complexity of a representation can lead to better domain understanding. Nonetheless, this counterintuitive result supports the relationships in the diagram shown in Figure 2-2. That is, representational simplification (and not representational complexity) is negatively related to domain understanding.

In a similar study, Parsons (2011) also empirically investigated the effect of explicitly representing property precedence on better domain understanding. His results suggest that increased complexity of domain representations correlates with better domain understanding. The result of this empirical study similarly supports the relationships shown in Figure 2-2 by confirming that representational simplification undermines domain understanding.

Although representational simplification is not recognised as an independent construct in the ontological framework of conceptual modeling, the discussions above show that its implications, to some extent, are known and addressed in this framework (both theoretically and empirically). In the search for an explanation of anomalies relating to the role of conceptual modeling, the ontological framework provides some theoretical guidelines (see Figure 2-2) and some empirical evidence.

Because the negative correlation between representational simplification and domain understanding has some support within the ontological framework of conceptual modeling, the framework may also provide insight about methods of conceptual modeling that do not rely on conventional techniques in representing domain semantics. In other words, while the ontological framework of conceptual modeling has been adopted extensively to formalise conventional practices of modeling (based on abstraction and generalisation techniques), it might also fit other modeling techniques that are not based on domain simplifications. However, this proposition requires further theoretical refinement as well as empirical support. For instance, the ways in which complex domains are better understood need to be determined if simplified representations do not guide the practice of system development. Furthermore, the concept of modeling and the differences between conventional and alternative practices of modeling need to be examined carefully.

As discussed earlier, this study explores the context of such differences between conventional and alternative practices of conceptual modeling. Henceforth, I first briefly discuss the literature on newly emerging information systems to compare different modes of simplification and methods used to improve domain understanding. I then discuss waterfall and agile methodologies to show the differences in the setting of the two system development methodologies.

2.1.6 Emerging New Information Systems and Alternative System Development Methodologies – Changes to Conventional Conceptual Modeling Methods

In newly emerging information systems, conceptual modeling paradigms and the underlying assumptions of information system representations are changing. As these systems increasingly are based on information obtained outside organisational boundaries, conventional approaches to conceptual modeling, mainly developed to represent domains within organisational boundaries, are becoming inadequate. For instance, in ubiquitous open systems, complete, *a priori* specifications of real-world domains no longer underpin modeling practices, nor are conceptual modeling practices based on the generalisation and abstraction methods used in class diagrams. This outcome arises because many emerging information systems focus on divergent and *unique* user views instead of *collective* of stakeholder views (Lukyanenko & Parsons, 2013a). In this regard, Lukyanenko and Parsons (2013a) argue that classification, as a prevailing mode of domain simplification, is deemed to lead to information loss. The reason is that classification is a consensus-driven approach, guided by the collective belief of stakeholders for *selective* representation of *relevant* information. The *selective* representation of information entails loss of information about domain semantics that are deemed to be irrelevant. In Table 2-1, I provide a comparison between different modes of simplification and their effects on domain understanding for different conceptual modeling methods.

Table 2-1 Overview of literature on context, method, modes of simplification, and domain understanding improvement in conceptual modeling

Conceptual Modeling Context	Conceptual Modeling Method	Mode of Simplification	Domain Understanding improved by	Reference
Conventional information systems —some general features discussed in Davis (1982) but no particular taxonomy is provided in literature	*A priori* requirements specification	Abstraction-driven classification	Representational fidelity	(Bodart, Patel, Sim, & Weber, 2001; Davis, 1982; Gemino & Wand, 2005; Parsons, 2011; Shanks, Nuredini, Moody, Tobin, & Weber, 2003; Wand & Weber, 1993, 1995)
Conventional information systems —no distinctive factor is provided in the literature specifying context suitable for agile methodologies	Lean modeling	High-level, barely complete representation	Iteration	(Ambler, 2005; Highsmith & Cockburn, 2001; Rajlich, 2006)
Distributed heterogeneous information with transient users and customisable features such as semantic search engines and social networking	No modeling	Instance-based representation	Accuracy per unit of data and greater number of instance acquisition	(Lukyanenko & Parsons, 2011b, 2013a, 2013b; Parsons & Wand, 2000)

As shown in Table 2-1, methods of conceptual modeling vary from complete *a priori* requirements specification in conventional practices to no modeling in emerging new information systems. While these approaches to modeling may reflect two extremes, there seems to be a middle approach to modeling—*viz*. lean modeling. In this form of lightweight modeling, which is common to agile methodologies, models minimally represent core phenomena in a domain. In this regard, the barely complete conceptual models of agile methodologies are adequate for the next design iteration. Highsmith and Cockburn (2001) contend that high-level, barely complete representations in agile modeling are a result of perceptions that uncertainty and change are inevitable in domain semantics. In agile methodologies, they argue that an evolutionary approach to lean modeling creates a feedback loop that improves domain understanding through iterations, thereby embracing uncertainty and change in a domain.

2.2 System Development Methodologies

As discussed earlier, my study explores conceptual modeling in traditional system development methodologies versus conceptual modeling in agile methodologies. In the following subsections, therefore, I provide an overview of traditional and agile system development methodologies. My objective is to delineate the differences between them and the assumptions that underpin them. I focus on only one form of traditional development methodologies—namely, waterfall methodologies—because this form exemplifies the characteristics of traditional methodologies.

2.2.1 Waterfall Methodologies

Waterfall methodologies were first introduced by Royce (1970) to develop information systems for large government projects. The process of developing systems using waterfall methodologies fundamentally resembles a manufacturing line. In waterfall methodologies, each phase of a project is completed fully before moving onto the next stage of development. Among the many different system development methodologies proposed since the late 1960s, the waterfall has been reported as a successful approach and one that is widely accepted (Huo, Verner, Zhu, & Babar, 2004).

System development based on the waterfall consists of five distinct stages: requirements analysis and definition, design, implementation and unit testing, integration and system testing, and operation and maintenance. The fundamental assumption that underpins waterfall methodologies is that specification of system requirements is possible in advance. Based on this understanding, the waterfall methodology is a linear process in the sense that specified system requirements are not revisited through a formal process. In other words, the recognition and necessity of *a priori* requirements specification obviates a need to formally revisit the requirements as projects develop or systems evolve.

Furthermore, in waterfall methodologies, practitioners' tasks in relation to each phase of the project are supposedly segregated. A team of analysts usually elicits and collates the system requirements. This team often moves onto another project when the requirements specifications are completed and released. The requirements specifications then provide the basis for the work of a team of system designers. Completed designs similarly are passed onto developers to implement program code and build the final software product. In this process, users or business stakeholders are not engaged continuously with the development project. Instead, they are expected to specify system requirements completely and in advance.

However, the disengagement of users and business stakeholders from the system development process impairs their vision about the final product in two ways. First, stakeholders do not have the opportunity to understand how the final system is achieved. Second, users and business stakeholders provide system requirements specifications without having a sense of how the final product will appear. Both effects arguably impact determination of system specifications adversely (Kovitz, 2003; Phatak, 2012; Royce, 1970; Szalvay, 2004). Without clearly understanding the phases of system development and without having a clear vision of the features in the final system, specifying domain requirements may become irrelevant. In other words, specification of domain requirements has a close relationship to how they are going to be achieved (development stages) and how they are going to address the problem space (features of the final product). The disengagement of stakeholders with either of these two elements can undermine the quality of system specifications.

Nonetheless, waterfall methodologies have a long tradition of wide adoption in information systems development. The acceptance of waterfall methodologies has occurred for many reasons. They are easy to adopt and follow, and stakeholders have a general familiarity with the waterfall. Also, heavy documentation, which is a core practice in the waterfall, enables stakeholders to follow the processes at any stage of the project. Furthermore, reliance on heavy documentation makes the waterfall robust with respect to formal evaluation processes through repeatability (Highsmith, 2002; Royce, 1970).

In spite of their wide acceptance, waterfall methodologies have been subject to substantial criticisms. These criticisms focus mainly on the waterfall's inability to respond to the inherent complexity and uncertainty in system development projects. Because the sequential perspective cannot cope with volatility and change, Szalvay (2004) argues the sequential perspective that underpins waterfall methodologies has effectively been abandoned in every other industry except software development projects. Furthermore, in spite of its popularity, many waterfall software development projects often go over budget and time and fail to meet user requirements (Erickson, Lyytinen, & Siau, 2005; Larman & Basili, 2003). Therefore, alternative system development methodologies have gained momentum.

2.2.2 Agile Methodologies

Agile methodologies have emerged as a result of organisations seeking to gain competitive advantage through timely development of information systems (Turk et al., 2005). Supposedly they are a paradigmatic and fundamental change to the sequential or lifecycle processes in the traditional methodologies (such as waterfall) (Rajlich, 2006). Agile methodologies are driven by concerns about the quality of developed systems (Ambler, 2005; Huo et al., 2004). With the evidence of 50% to 75% failure rates in traditional system development projects, and over half of operational and maintenance budgets compromised for inadequate system analysis (Erickson et al., 2005), a strong case for adopting alternative development methodologies, such as agile, exists.

Two other concerns influence the emergence of agile methodologies. One is the need for creativity. The other is the need to build workplaces that are more favourable to people. Szalvay (2004) argues that a culture of experimentation and learning is promoted when agile, iterative approaches, and incremental change is used. Creativity and innovation arise by breaking the status quo.

In spite of their alleged advantages, agile methodologies have also been subject to criticisms. Rajlich (2006) considers agile methodologies to be transiting through their infancy. Lee and Xia (2010) argue that agile methodologies are based primarily on rhetorical and anecdotal arguments; there is no clear understanding based on empirical studies that examine the effects of agile practices. Boehm (2002) dismisses the claims of universality with respect to agile methodologies and proposes that both agile and waterfall or hybrid methodologies are relevant to different types of systems, depending on the risks involved with each system project. Cockburn (2002) describes agility as difficult to execute in practice.

Nonetheless, agility is considered to be a dynamic, context-specific, and change-oriented methodology (Highsmith & Cockburn, 2001). Agile methodologies emphasise informal stakeholder communication and iterative processes in system development (Ambler, 2005; Conboy, 2009; Lee & Xia, 2010; Turk et al., 2005). Quality in agile methodologies is achieved through rapid delivery of software, so that stakeholders' feedback can be obtained earlier. Simplicity of the solutions is emphasised, so that the cost of change is reduced. Continuously improving the design is stressed, so that future implementation stages become more straightforward. Finally, continuous testing is emphasised, so that defects can be identified earlier at lower cost (Highsmith & Cockburn, 2001).

In defining agility, Conboy (2009) argues that an information system development methodology is considered to be agile if any of the processes in the methodology subscribe to change, in at least one form of creating change, reacting to change, learning from change, or being proactive toward change. Additionally, he argues that an agile methodology "must not detract from any of the perceived economy, perceived quality, and perceived simplicity" (Conboy, 2009, p. 341). Last, he argues that any distinct part of an agile methodology must be always readily available. This means that the time and the costs of agile-related processes must be minimal.

Practitioners in agile methodologies, rather than researchers, are mostly driving creation of the body of knowledge (Abrahamsson, Warsta, Siponen, & Ronkainen, May 2003; Conboy, 2009; Lee & Xia, 2010). In the Agile Manifesto (Beck et al., 2001), which is the canonical document since inception of this methodology, practitioners declare:

> *"[We] value,*
>
> *Individuals and interactions over processes and tools*
>
> *Working software over comprehensive documentation*
>
> *Customer collaboration over contract negotiation*
>
> *Responding to change over following a plan."*

In the light of the four core values of the Agile Manifesto, different agile methodologies including XP, Scrum, and the Crystal family have been developed (Ambler, 2005). Adoption of agile methodologies is based on using specific techniques, such as pair programming and time-boxed and test-first development practices (Thummadi, Shiv, Berente, & Lyytinen, 2011). Nonetheless, each of these principles has clear implications for practice. For instance, preference of people over processes motivates collaborative, self-organising teams that do not invest in heavy documentation in project developments. Similarly, a preference for customer collaboration over contract negotiation promotes an evolutionary approach to development of systems among business and technical stakeholders. In the following subsections, I discuss the assumptions that underpin agile methodologies in the context of these four core values.

2.2.3 Assumptions Underpinning Agile

Preference of individuals and interactions, over processes and tools

Contrary to the traditional perspective on system development projects, agile methodologies position the practice of system development as a “decidedly human activity” (Highsmith, 2002, p. 6). From an agile perspective, this means that *people* rather than *processes* are the primary drivers of success in system development projects. From this perspective, no processes, however complete and clear they are, can compensate for the talent and skills of people involved in the project. As a result of this belief, team collaboration, creativity, and self-organisation constitute core practices in agile methodologies. The reliance is on team diversity and team autonomy to facilitate individuals’ efficient and effective decision making. The insistence on individuals’ decision making in agile methodologies contrasts clearly with waterfall practices where pre-determined sets of rules as guidelines for decision-making are heavily emphasised (Lee & Xia, 2010).

Furthermore, preferring people to processes and tools leads to decreased reliance on heavy documentation in agile methodologies. Instead, *lean documentation* is preferred to guide practice in agile. Compared to the formal documentation and modeling tools used in traditional methodologies, people are deemed to be a faster and better means of communication in agile.

Ultimately, agile methodologies have a preference for people interaction over documentation and tools, especially in complex information systems development contexts. As the level of complexity rises, more enriched mediums of communication, such as face-to-face interactions, are deemed necessary if information systems projects are to succeed (Highsmith & Cockburn, 2001).

Preference for working software over comprehensive documentation

In agile methodologies, efficiency is defined through *simplicity*. This means that developing software, which answers current stakeholders' needs, is more efficient than coming up with solutions that can be used over time. On this basis, agile methodologies emphasise *lean modeling* to capture informal design specifications.

These specifications are given in the form of informal models on presentation mediums such as whiteboards or sketch papers. The objective of these presentations is to communicate the ideas with no intention of precision (Turk et al., 2005). These ideas often depict high-level subject matter with barely complete specifications.

Because working software is preferred over formal and extensive documentation, the notion of "unforgiving honesty of a working code" (Highsmith, 2002, p. 20) is emphasised in agile methodologies. That is, in agile methodologies, working code, and not modeling scripts, is the primary goal (Ambler, 2014; Highsmith, 2002; Highsmith & Cockburn, 2001).

Preference for customer collaboration over contract negotiation

Highsmith and Cockburn (2001) argue that the quality of the written code in agile methodologies is determined by the *behavior* of human agents who write the code. Similarly, Sumrell (2007) reports how a collaborative approach in an agile project improves the quality of the developed system. In agile methodologies, every project team member is responsible for the quality of the developed system, not just the team that is in charge of quality assurance.

In an ethnographic study on agile practice, Sharp and Robinson (2004) describe how a team of agile practitioners collaborated on-site and face-to-face with their customers in a way that the final system was the result of a shared responsibility between customers and developers team (as opposed to delegating responsibility to the developer's team based on a negotiated contract). Although working with customers is complicated, Ambler (2005) argues it is beneficial, because no stakeholder is capable of specifying all the requirements. He therefore asserts that business and technical stakeholders must work together *daily* for the period of the project to maintain a constant pace in developing the system.

With a tendency among agile practitioners to become "generalising specialists" (Ambler, 2005, p. 39), customer collaboration has become an important characteristic of agile projects. In Ambler's (2005) view, the importance of becoming a generalising specialist lies in the ability of practitioners to interact effectively with others (including customers). In the context of an evolutionary approach to system development that is practiced in agile methodologies, Ambler (2005) explains how agile practitioners hold a general understanding of the development processes and the domain. This general understanding, as opposed to a narrow specialisation, enables effective engagement of all stakeholders in a collaborative approach. Such collaboration in agile methodologies is replacing contract negotiation as a top-down delegation approach to specialists in traditional methodologies.

Preference for responding to change over following a plan

One of the fundamental differences between agile and waterfall methodology is in the way they perceive change in real-world domains. Waterfall methodologies assume that *a priori* specification of a complete set of domain requirements is possible. Therefore, any change to a requirements specification after they have been finalised is deemed to contribute to system failure. In agile methodologies, however, freezing requirements at an early stage of system development is perceived to contribute to system failure because the development process becomes *unresponsive* to business needs.

Therefore, agile methodologies seek to minimise the cost of change during development processes rather than to avoid it (Highsmith & Cockburn, 2001). As a consequence, *incremental change* based on iterations through an evolutionary approach to system development constitutes a fundamental method in agile practices (Rajlich, 2006). By perceiving change as an inevitable element in real-world domains and embracing it, agile methodologies adopt change as shaping their development methods through an iterative approach to system development.

To describe the element of change in the context of agile practices, Highsmith (2002) uses an analogy between a battlefield and a system development project. Although extensive planning in system development projects is important to success, similar to the battlefields, he argues that success is not assured by following a detailed plan. Instead, as winning in the battlefield is characterised by probing the enemy and responding to their actions, projects also require recognising change and responding to it.

Furthermore, although system development projects have relatively clear objectives, Highsmith (2002) argues that the specific system requirements are volatile and evolving. He identifies *volatility* of system requirements to be a characteristic of the context of practice. In his view, a volatile context demonstrates *"*high-exploration factors [not succumbing] to rigorous [and] plan-driven methods*"* (Highsmith, 2002, p. 4). He asserts volatility occurs because project stakeholders embark on exploring unknown domains. In his view, *uncertainty* arising from the absence of complete knowledge about domains also characterises system development projects that fit agile methodologies.

However, he also argues that agile methodologies are not limited in their application to volatile systems. Instead, he maintains that every system project context is chaordic—that is, no system project is predominantly ordered or chaotic. Instead, it is a combination of both. For this reason, he argues that agile is not a specific methodology that fits only highly uncertain domains. Instead, agile is an *ecosystem* that provides a chaordic perspective on system development projects through collaboration and barely sufficient modeling as its underpinning assumptions (Highsmith, 2002).

In summary, Table 2-2 provides an outline of the implications of agile's assumptions for practice.

Table 2-2 Implications of agile assumptions for practice

Agile assumptions	**Implications for practice**
People preferred over processes	Creative, collaborative, and self-organising teams Lean documentation
Working software preferred over comprehensive documentation	Simple solutions Lean modeling as communication tools with no intention of precision
Customer collaboration preferred over contract negotiation	An evolutionary approach to system development
Change preferred over following a plan	Incremental change through iteration Volatility and domain uncertainty as context of practice Evolutionary approach

To gain a better understanding of the role of conceptual modeling in alternative methodologies, it is important to note the resemblance between Highsmith's (2002) battlefield analogy and the interpretation provided earlier by Lukyanenko and Parsons' (2013a). Highsmith (2002) advocates a probe-and-sense mechanism to better understand domains in agile methodologies. Similarly, Lukyanenko and Parsons (2013a) propose that conceptual models are regarded as sensitising tools to obtain more information about the real-world domains in alternative systems.

Although further research must be undertaken to understand the nature of probe-and-sense mechanisms as well as modeling as a sensitising tool, the convergence of these interpretations with what is traditionally understood to be the role of conceptual models in improving domain understanding is promising. Compared to traditional methodologies, the practice of conceptual modeling and system development may have changed drastically. Nonetheless, a shared assumption exists about the role of conceptual models in improving domain understanding. This assumption provides a basis to understand alternative methodologies through the theoretical lens of the ontological framework to conceptual modeling.

2.3 Summary

This chapter has addressed two topics. First, I presented the theoretical framework that underpins my research. Specifically, I discussed the Ontological Perspective on Conceptual Modeling as a theoretical lens that can be used to understand the practice of conceptual modeling in agile methodologies. A definition of conceptual modeling was provided, followed by an explanation of the concepts of conceptual modeling scripts, grammars, methods, and contexts. I then showed how each of these elements forms the basis for the ontological research program on conceptual modeling practice. Furthermore, the objectives of conceptual modeling practice were discussed, and arguments were presented on the relationship of conceptual modeling representational fidelity and the success of information systems.

In the context of the ontological research framework to conceptual modeling, I highlighted the under-researched areas of conceptual modeling methods, context, and the interplay between methods and context (Figure 2-1). In addition, I discussed a competing explanation on the role of conceptual modeling practice in improving domain understanding (Figure 2-2). I presented this discussion to problematise the theoretical assumptions that are conventionally held about the role of conceptual modeling in system development projects. Although representational fidelity could improve domain understanding, the problematisation indicated that an element of representational simplification exists during the practice of conceptual modeling that could potentially undermine domain understanding based on the represented models.

The discussed problematisation of the conceptual modeling assumptions was presented against a background that is already under scrutiny by alternative system development methodologies and emerging new information system. I provided a brief review of literature on alternative systems and their assumptions.

I then discussed the nature of waterfall and agile methodologies. This overview provides the basis for addressing the proposed research questions in Chapter 1. In an interpretive study, understanding the underpinning assumptions of system development practices is essential to explaining the perceived anomalies in the role of conceptual models.

Most importantly, as a result of this overview, I identified a convergence among different perspectives to conceptual modeling role. In spite of different understandings about the role of conceptual modeling in traditional and alternative systems and methodologies, the overview of the literature highlighted that a consensus exists about their role. Conceptual models in alternative systems and methodologies are regarded as *probe-and-sense* or *sensitising tools* that are devised to improve domain understanding. Domain understanding is also the focus in traditional methodologies. In spite of presence of different understandings about conceptual modeling role in traditional and alternative system development methodologies, a better understanding is needed, however, of how other elements in the context of practice impact conceptual modeling methods as a means of obtaining better domain understanding. In this regard, the focus of my research is to understand the under-researched area of how methods and the context of information systems interplay to gain a better understanding of real-world domains.

3 RESEARCH METHODOLOGY

This chapter describes the methodology and the conduct of my research. Myers (2013) stresses that qualitative research is comprised of five building blocks: research philosophical assumptions, research method, data collection technique, data analysis approach, and a written record of findings. In the sections below, I elaborate on each of these building blocks, specifically in relation to my study, before I present my findings in the next chapter.

First, I briefly discuss a qualitative research design based on Myers (2013) and the interpretive paradigm that underpins this design. Second, I describe the semi-structured interviews that I used as a data collection method. Last, I discuss hermeneutics as the interpretive approach I used for data analysis, and thematic analysis, as the specific data analysis method that I employed.

3.1 Interpretive Field Study Research – The Rationale

This section describes the rationale for using an interpretive field study to undertake this research.

3.1.1 Qualitative Research Design

Myers (2013) defines a qualitative research design as a method that provides researchers with the ability to observe and understand the context of humans' practices. Through this understanding, qualitative research can provide insights in relation to *what* is taking place, the *reasons* behind the formation of certain practices, *why* certain practices are significant, and *how* a particular practice becomes relevant in a specific context.

As discussed in Chapter 2, the practice of conceptual modeling has been researched extensively with respect to the quality of its modeling grammars and scripts. The use of conceptual modeling methods to elicit system requirements in the context of system development methodologies and the interplay of these methods and context, however, are less explored. For this reason, I argue that a qualitative research design enables me to observe and explore the context in which the real-life practices of system development, including conceptual modeling, take place. Through this exploratory study, I seek an understanding of *what* factors are informing human decisions about the practice of conceptual modeling, *how* these decisions are manifested in practice, and *why* these factors are relevant in the context of practice.

3.1.2 Interpretive Philosophical Assumptions

In the spirit of interpretive research (Kaplan & Maxwell, 2005), the focus of this study is to understand the complexity of the context of conceptual modeling practice, as described by practitioners. The extant literature on system development methodologies, system requirements elicitation, and the ontological perspective on conceptual modeling inform our theoretical understanding of conceptual modeling practice. This theoretical understanding also guides the scope of this study and the research questions addressed.

On the other hand, interpretive research on the practice of conceptual modeling in agile methodologies contributes to an in-depth understanding of the phenomena of interest through the meaning practitioners assign to it (Orlikowski & Baroudi, 1991). The theoretical insights obtained from the literature assist with the interpretation of the collected data (Klein & Myers, 1999). These theoretical insights are not used deductively to impose hypotheses on the collected data (Myers, 2013). Rather, the data is analysed to provide an understanding of the practice, meaning, and intentions of practitioners.

3.1.3 Interpretive Field Study

Interpretive field study research includes in-depth case studies and ethnographies (Klein & Myers, 1999; Walsham, 1995). Yin (2014) distinguishes between ethnography and case study research based on the required time for the research and the researcher's level of immersion in the field. While ethnographies often expand into several months of in-depth learning from people through observations and fieldwork, case studies usually takes some weeks of studying research participants, mainly through interviews. Case study research is conducted to demonstrate whether a particular theory or proposition is applicable in a specific context (Benbasat, Goldstein, & Mead, 1987; Myers, 2013).

Case studies can be exploratory (Yin, 2014) or explanatory (Myers, 2013). Irrespective of the underlying philosophical assumptions, Myers (2013) defines the idea of a *case* as describing a particular situation through its more general characteristics. Case study research is therefore characterised as "[using] empirical evidence [mostly from interviews and documents] from one or more organisations when an attempt is made to study the subject matter in context" (Myers, 2013, p. 76). Creswell (2012) defines case study research as exploring an issue within a *bounded system*, such as a setting.

Based on these definitions, the case, the issue, or the subject matter in this study is the practice of conceptual modeling, while the context is agile system development methodologies. Because this definition of case study is neutral to the underpinning philosophical assumptions (Myers, 2013), sets of guidelines for interpretive field study research, which includes case studies (Klein & Myers, 1999; Walsham, 2006), inform the conduct of this research.

3.2 Semi-Structured Interviews

This section describes how I used semi-structured interviews as the data collection method in my research.

3.2.1 Data Collection – Method

Interviews are often the most-important sources of data in qualitative research. They allow researchers to obtain participants' understanding about the phenomena of interest (Myers, 2013; Walsham, 1995). Creswell (2012) identifies interviews as one of the four main forms of qualitative data collection (the other three are observations, documents, and audiovisual materials). Three types of interviews can be used: structured, semi-structured, and unstructured (Myers, 2013). Structured interviews are based on complete pre-formulation of the questions and the setting of the interview.

Unstructured interviews, on the other hand, are not based on predetermined sets of questions or the setting.

By analogy to a drama setting (Myers & Newman, 2007), semi-structured interviews are guided by *improvisation*. While structured interviews serve the purpose of consistency across all interviews and unstructured interviews allow participants' freedom, semi-structured interviews are the most-common method of data collection. They are preferred because they balance the disadvantages of the former two. For this reason, I used semi-structured interviews as the data collection method in my study.

3.2.2 Data Collection – Conduct

Creswell (2012) indicates that the conduct of interviews involves seven steps. In the first step, a researcher identifies participants based on a purposeful sampling strategy (Marshall, 1996). The initial sampling strategy used in this research is the snowball or chain strategy (Miles & Huberman, 1994). The snowball strategy was initiated by a referral to conduct a pilot interview with a prominent practitioner in the field. The pilot interview had the purpose of evaluating the feasibility of conducting a case study project based on its appeal to the community of practitioners. Furthermore, the results of the pilot interview guided reframing of the research questions according to the modeling practices that were used within the community of practitioners. Upon completion of the pilot interview, I revised the initially drafted research Explanatory Statement in light of the insights gained from the pilot interview. I then used the revised Explanatory Statement and the Consent Form (Appendix A) as a point of contact to recruit future participants. In addition, I developed a protocol for conducting semi-structured interviews (Appendix B).

To collect data, I interviewed eight male practitioners who worked in five different industries—namely, health, insurance, consultancy, gaming, and telecommunications. The practitioners were highly respected professionals in the field of information systems development and design. They had between 5 and over 20 years of experience in Australia and overseas. Except for one teleconference interview with a highly recognised practitioner based overseas, all interviews were conducted face-to-face in an office location chosen by the practitioners. I conducted over 11 hours of interviews with the practitioners, including a pilot interview. Except for the pilot interview, I interviewed each of the practitioners once for approximately one hour. The pilot interview took just more than two hours and the pilot interviewee agreed to participate in the main series of the interviews, sharing his insight further about the revisited research questions. As a result, this practitioner was interviewed twice.

With the consent of the practitioners, all interviews were recorded on two electronic devices (so that a backup recording was always available). All interviews were then fully transcribed. The transcriptions of the interviews constituted over 40 pages of A4 size and more than 32000 words. Some of the practitioners provided some sample modeling scripts to assist my discussions with them. Two of the practitioners also used a whiteboard to demonstrate their points. With their consent, I took photos of the whiteboard materials to aid transcription of the interviews.

Table 3-1 provides a summary of how I conducted the interviews and some demographic information about the participants. Although the initial recruiting strategy in this research was snowball, Table 3-1 shows that I also used other strategies such as random purposeful as the data-collection phase progressed. As a result, I was able to interview practitioners from multiple sites, industries, and regions. This variation enhanced the validity of the collected data by decreasing data context-dependency (Creswell, 2012; Newman, 1998).

Table 3-1 Details of method conduct

Informant Index	Collective Experience in System Development including Modeling in Agile	Sampling Strategy	Form of the Qualitative Data	Maximum variation by			Recruited by
				Industry	Site	Region	
Practitioner 1	Over 20 years	Snowball	Interview / documents	x (Health)			Referral
Practitioner 2	Over 10 years including overseas	Snowball	Interview / documents	x (Health)			Referral
Practitioner 3	Over 5 years	Snowball	Interview / documents	x (Health)			Referral
Practitioner 4	Over 10 years	Random purposeful	Interview	x (Gaming)	x		Direct approach - Expression of interest
Practitioner 5	Over 20 years	Random purposeful	Interview	x (Insurance)	x		Direct approach - Expression of interest
Practitioner 6	Over 5 years	Random purposeful	Interview	x (Telecom)	x		Direct approach - Expression of interest
Practitioner 7	Over 20 years including overseas	Random purposeful	Interview / documents	x (Consultancy)	x		Direct approach - Expression of interest
Practitioner 8	Over 20 years - overseas	Politically important	Interview	x (Consultancy)	x	x	Public profile – Direct approach

Among the eight practitioners I interviewed, three were initially working on the same project at a single site. Once I completed the interviews with these three practitioners, I obtained a referral to attend an internationally recognised conference by an association of practitioners in the field of modeling. Upon attendance, the chair of the association introduced my research project to participants and sought expressions of interest from the audience to take part in my research. By personally attending the conference, I also gained the opportunity to meet and discuss the purpose of the research briefly with the interested practitioners. Four participants from four different industries were ultimately recruited from the association conference meeting. In addition, I contacted a number of high-profile agile practitioners, globally, by accessing their public profiles. Among those approached, one *politically important* agile practitioner (Miles & Huberman, 1994) who was based overseas agreed to be interviewed.

Besides data I obtained from the interviews, I asked all practitioners if they could provide any modeling materials that they used in their practice. Four practitioners provided samples of modeling scripts that they used at the time in their projects. This documentation was also considered alongside the interview transcripts for analysis and interpretation of the collected data.

3.3 Hermeneutics - The Interpretive Approach to Data Analysis

As discussed in Chapter 2, the quality of conceptual models is believed to determine information systems success in traditional system development methodologies. In alternative methodologies such as agile, however, no formal recognition is given to the role for conceptual models. Furthermore, as shown in Fig. 2-2, there seems to be a competing explanation for the role of conceptual models in relation to better domain understanding. Based on this competing explanation, while high-quality conceptual models purportedly enhance domain understanding through their representational fidelity, they also seem to be obfuscating domain understanding by simplifying the semantic complexity of the corresponding domain. Thus, a contradiction exists in relation to the role of conceptual modeling.

In such situations, Myers (2013) discusses how *hermeneutics* provides a set of concepts to enable in-depth understanding of qualitative data, especially where contradictory interpretations of situations exist. By focusing on the *meaning* of the qualitative data, hermeneutics helps to clarify the nature of human actions and the reasons behind the actions.

Hermeneutics can be used in two different ways. It can either be used as a mode of analysis (Myers, 2013), or it can inform the assumptions that underlie interpretivism (Klein & Myers, 1999). In this study, hermeneutics is applied as the interpretive approach to data analysis. Sarker and Lee's (2006) insights have particularly guided the way in which I have adopted hermeneutics in my study. Accordingly, I adopted the *fundamental principle of the hermeneutic circle* (Klein & Myers, 1999) to try to obtain a coherent understanding of the contradictory interpretations of conceptual modeling practices in the waterfall and agile contexts.

The *fundamental principle of the hermeneutic circle* (Klein & Myers, 1999) emphasises the iterative movement between the parts and the whole of a text in order to gain a coherent understanding of the complex context the text is describing. Through this iterative movement, the meaning of the parts is understood with respect to the whole, and the meaning of the whole is understood with respect to the meaning of the parts and their interrelationships.

In this study, I understood and *coded* the chunks of qualitative data from interview transcripts with respect to the meaning of each single interview as a whole. Subsequently, I revised the meaning of each coded part and sought to understand it in relation to the set of all interviews. Moreover, I interpreted each chunk of data in the context of a global understanding based on literature and other data. Figure 3-1 shows chunks of data as *parts* versus a complete interview and sets of interviews as *wholes*.

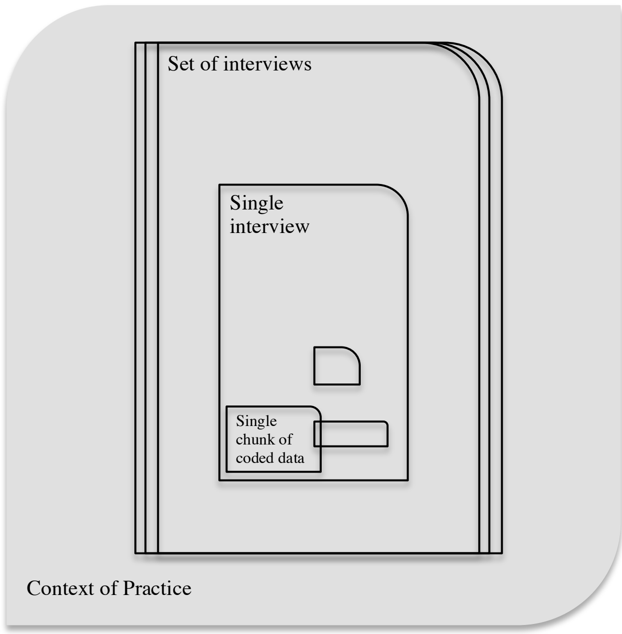

Figure 3-1 Representing data parts embedded in their respective whole as a notion based on the Principle of the Hermeneutic Circle

3.4 Thematic Analysis – The Data Analysis Method

As shown in Fig. 3.1, chunks of *coded* data constitute the first-order parts in the hermeneutic circle of interpretations (Miles & Huberman, 1994, p. 56). However, there are different methods of coding used for analysis of data in qualitative studies (Myers, 2013). In this study, the method used is *Thematic Analysis* (Braun & Clarke, 2006; Thomas & Harden, 2008). I chose this method because it can accommodate the theoretical and epistemological framing of the research (Braun & Clarke, 2006); thus, it is suitable for interpretive field studies. Additionally, I chose thematic analysis because it is widely used as a foundational method across major qualitative traditions. While applied effectively as a method for data reduction, thematic analysis organises and reports on the collected data in rich details (Braun & Clarke, 2006).

3.4.1 Phases of Data Analysis

Braun and Clarke (2006) identify six phases of thematic data analysis: familiarisation with data, generating initial codes, searching, reviewing, defining and naming of themes, and producing a report. In this study, I transcribed all interviews. I then reviewed and checked the recording against the transcription to fulfill the familiarisation phase of data analysis. Furthermore, I wrote memos about the initial ideas after I conducted and transcribed each interview. These memos assisted with the process of coding, subsequent to the completion of the initial phase. Figure 3-2 shows a snapshot of the coding environment. The coding was done using NVivo.

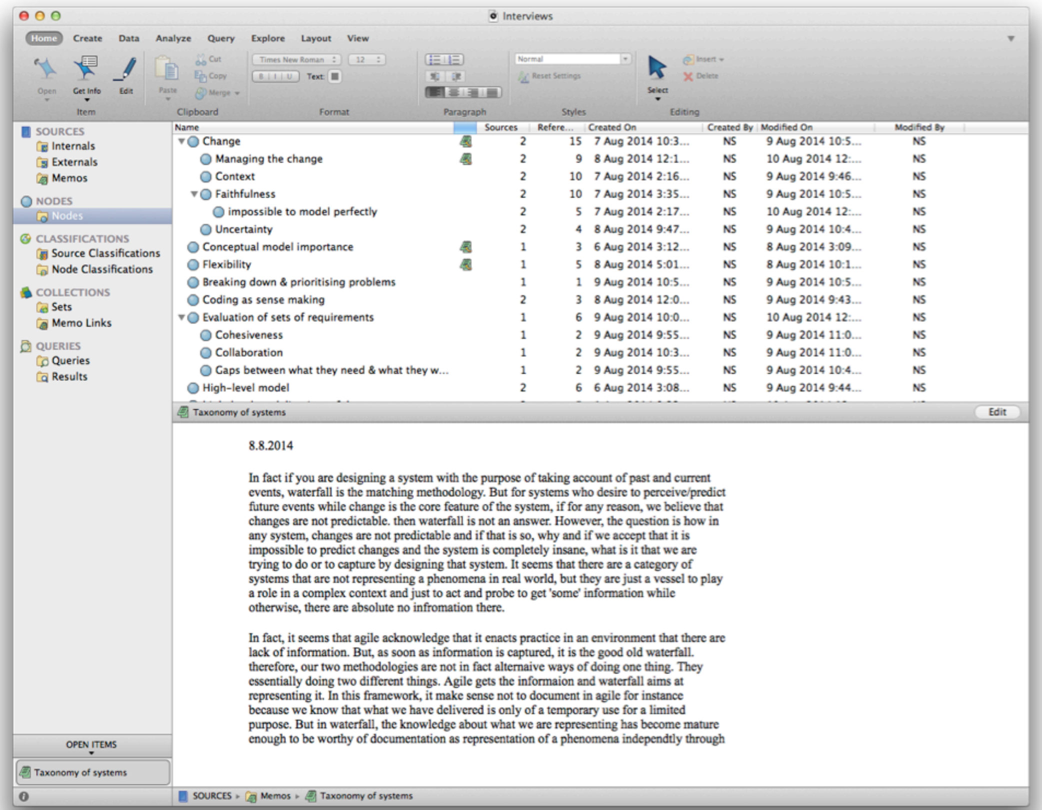

Figure 3-2 A sample snapshot of the coding environment in NVivo

These codes, by definition (Braun & Clarke, 2006), referred to the most basic elements of the data that made sense in terms of the practice of conceptual modeling or the context of agile system development methodologies.

3.4.2 Identification of themes

According to Braun and Clarke, "a theme captures something important about the data in relation to the research question, and represents some level of *patterned* response or meaning within the dataset" (Braun & Clarke, 2006, p. 82). I identified some themes in this study based on the repetition of their underlying codes in the data. Nonetheless, code quantification has not been the only guide to theme identification. I identified some themes based on the significance of the concept they denoted in relation to the research question. Furthermore, an inductive analysis of the data and the codes informed identification of the themes in this study.

As the purpose of this study is to explore the context and the practice of conceptual modeling, I did not impose any pre-existing theoretical framework on the creation of the themes. Instead, I adopted a data-driven thematic analysis (Braun & Clarke, 2006) to allow the themes to emerge from the dataset. Also, I identified themes in the *semantic layer* (Braun & Clarke, 2006) based on their explicit and surface meaning. In this regard, no underlying assumptions were examined at data level. Instead, I captured a descriptive narration of data based on the explicit meaning of the data. I then organised the codes based on the emergent patterns in the dataset and significance of the notion they conveyed.

Ultimately, through adopting hermeneutics as my data analysis approach, I *interpreted* themes for their broader meaning and their implications in relation to my research questions and the literature (Braun & Clarke, 2006). In the next chapter, I present a detailed description of how I executed the data analysis method. I also present the *findings* of research of my research.

3.5 Summary

This chapter described the first four building blocks of a qualitative study before I present the fifth building block, the *findings*, in the following chapter.

I first explained why an exploratory study suits the purpose of this research through illuminating the factors that impact human decisions in the practice of conceptual modeling. I then explained why I adopted an interpretive field study to facilitate an understanding of how and why these factors are relevant in the interplay of conceptual modeling methods and context. I also provided a detailed explanation of the method of data collection, including recruiting eight practitioners who had extensive experience in system development projects, and the conduct the semi-structured interviews. Last, I described hermeneutics as the data analysis approach and thematic analysis as the method of data analysis to set out the context for presenting the *findings* in the next chapter.

4 FINDINGS

This chapter presents the findings of my study. A detailed description of how the interviews were conducted is followed by a report on the thematic analysis of the dataset. The thematic analysis uses data extracts to demonstrate how themes are identified. In the subsequent sections, I describe each theme and its relevant subthemes in detail as a lens to reflect on the interviews and to provide a cohesive interpretation of data.

4.1 Thematic Analysis of Data

As discussed in Chapter 1, a major objective of this exploratory study is to evaluate the extent and significance of conceptual modeling in the practice of agile methodologies. In this regard, understanding practitioners' views about the need for and the importance of conceptual modeling is essential.

As discussed in Chapter 2, a second and equally important purpose of this study is to gain insight about the settings in which agile methodologies are used. Gaining such an insight is important because it essentially addresses the need for better understanding of the unexplored areas of method, context, and their interplay. Indeed, gaining such an insight is in line with Wand and Weber's argument that "we need to understand better how the context affects modeling work and, in turn, how modeling and use of models affect elements of the context" (Wand & Weber, 2002, p. 371).

Because I adopted semi-structured interviews for data collection, the introductory interview questions (Appendix B) were followed by probing questions to enable exploration of the contextual elements of modeling practices. Table 4-1 provides an excerpt from an interview, which shows how probing questions were used to explore the underlying elements of modeling practice.

Table 4-1 Excerpt from an interview showing exploration of contextual elements

Researcher	Do you use conceptual modeling in your practice, and how do you evaluate the significance of such practice if it is used at all?	Introductory Questions
Practitioner 5	Particularly here, [in this industry…] it has been very useful for me to put a conceptual high-level logical structure around what we are doing […] Because, if the business is now starting to think ok, information is an asset, how are we going to manage it rather than to think we need to capture all that transactions; it starts thinking about information as information not as transactions associated with doing business. So, what I think is happened is that the whole conceptual piece of modeling, in its theoretical positions were valid to business for the past […but] actually [it] has risen from an almost theoretical university type of stuff that is just good as contemplation type of stuff, it actually has real value to the organisations now.	
Researcher	Can you perhaps elaborate in your experience, what importance the notations have in this process?	Probing Question
Practitioner 5	In technical form, it is absolutely critical. [Because] you have got to know absolutely what is optional and what is mandatory because you have got these foreign keys, and one-to-many and many-to-many relationships. And, it is the thing that drives you nuts when you are doing warehousing and information management delivery type of stuff.	
Researcher	Why? Is it because it is not consistent across the board or is it because…	Probing Question
Practitioner 5	Because as I said, the fact that it ***changes***, what I was told upfront is no longer true; it is true in this system but not true in that system. The code system which is mandatory here is not the code systems which is mandatory there; they code different things and so, you have got this massive exercise because you have received fairly ***rigid structures*** in they are strict with third normal form, foreign keys, big complex records that you do not want to change.	Identification of two contextual elements as ***change*** and ***rigid structure***

I coded the content of the entire dataset, which includes all the interview transcriptions. The initial list had 57 codes. Table 4-2 shows the initially generated codes with some samples of the associated data extracts.

Table 4-2 Initially generated codes and some associated data

Initially Generated Codes	**Sample Codes and Associated Data Extracts**
Agile versus Waterfall Change Classification Coding as sense making Cognitive engagement Cohesiveness Collaboration Common understanding Communication Complexity Conceptual model importance Conceptual modeling being remote Conceptual modeling being unrealistic Conceptual modeling being too configurable Conceptual modeling depends on audience – no single model is possible Conceptual modeling depends on purpose – no single model is possible Context Discordant view Documentation Documentation versus delivering functionality Faithfulness Focus Evaluation Functionality Flexibility Gaps between what they need and what they want High-level versus low-level models High-level models' usefulness High-level models' importance Impossible to model perfectly Information as asset IT industry Incremental design Industry relevance Lack of clarity Lack of continuum in stakeholders Limited experience Managing the change Modeling practices are diverse Models are fragmented—no single model possible Ownership Requirement breaking down and prioritising Resolving fuzziness Representation Scoping Stakeholders' organisational positions Stakeholders Standardisation Support in agile Systems failures Taxonomy of systems Technology and conceptual modeling Theoretical support in modeling Translation Uncertainty Value Various perspectives	***Standardisation*** "Normally they say there are multiple ways of representing conceptual models. This is based on relational schema. Then, we have UML. There are multiple ways. But unfortunately because there are no standard professional requirements that everybody should read the language, it is limiting the ability of these interactions." ***Complexity*** "Most of the projects I had been involved in…. There had been a little bit of effort put into getting the bigger picture. Like any profession, there has been gaps and deficiencies and things like the scenario of the "forgotten bathroom" we call it, but it can and has come about the lack of understanding of the requirements or clarity or sometimes the business does not know what they want until they put it into some context…." ***Value*** "We don't need to model it in such detail. This is actually a very expensive square. The function business model intersection is really expensive to do."

4.1.1 Searching for Themes

As coding of the dataset proceeded, some patterns in relation to the underlying meaning of the dataset began to emerge. Patterns mainly represented the frequency of the identified codes. For instance, responses that indicated the notion and therefore the coding of *'complexity'* formed the second most-frequent pattern after those that were collated under a very broad coding of *'agile versus waterfall.'* Therefore, *'complexity'* was recognised as a theme, capturing a seemingly important concept about the research question based on the frequency of appearance.

On the other hand, a few codes formed themes, not based on their frequency, but due to the significance of their meaning. For example, *'discordant view'* was coded in relation to only one practitioner's comment in a limited context. Nonetheless, *'discordant view'* was preserved as a theme because of its importance in relation to another two themes—namely '*flexibility'* and *'transcended problem-solving method.'*

In the search for potential themes, I reanalysed each of the codes shown in Table 4-2 with respect to other codes, to identify possible inter-relationships. I revisited each code-related data extract. While some data extracts were recoded as a result of this process, some codes were found with no particular relationship to other codes. Figure 4-1 shows an excerpt of an initial thematic map for a selection of codes and themes. I developed a complete thematic map that includes all codes in Table 4-2 before undertaking revision of the themes.

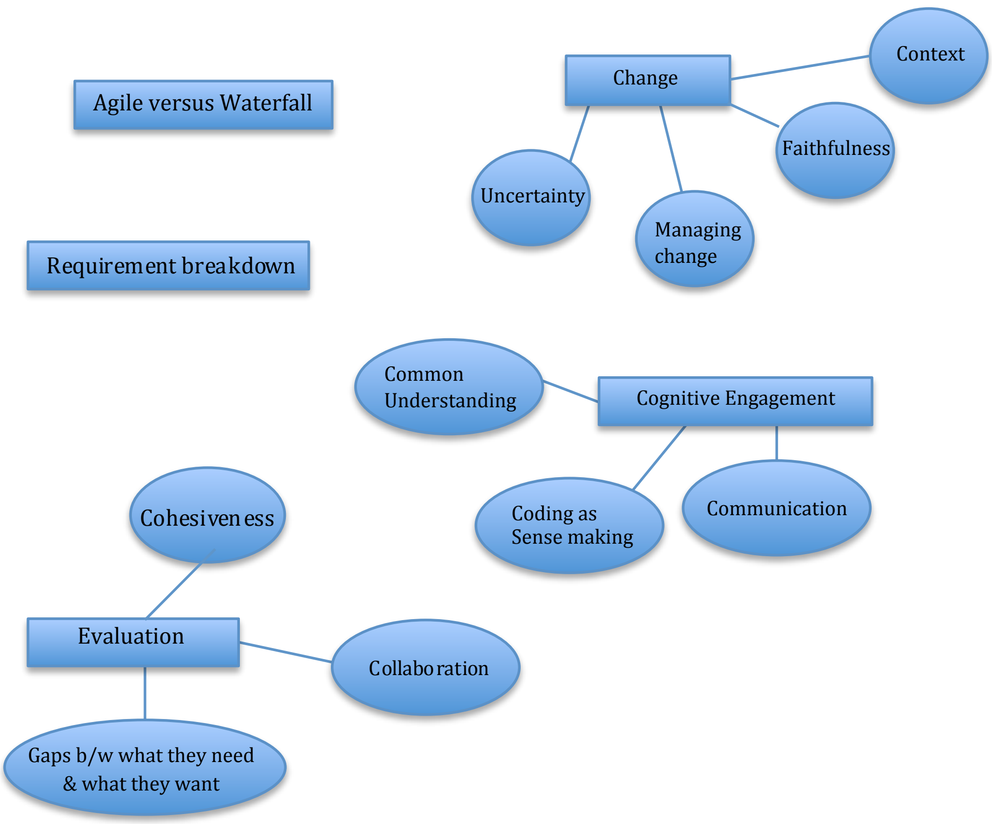

Figure 4-1 An excerpt of an initial thematic map showing some code-theme configuration

4.1.2 Revision of Themes

As searching to identify themes and refining the thematic map of data progressed, I observed that the data I obtained does not provide equal support for all the candidate themes. Some candidate themes had only a few data extracts to underpin their proposed meaning. Others had data extracts that were too diverse to extract a coherent higher-level meaning. I discuss refinement of each of these categories in the following subsections.

Not reaching data saturation - As indicated above, some candidate themes subsequently had few data extracts to support them. For instance, I identified no subthemes associated with the initial candidate theme of *'support in agile.'* In this regard, I coded only a single data extract for this theme. Practitioner 6 had commented:

> *"The thing is that when we had to do a second go for the same product, we did improve the functionality for the same product and we said, ok, what were the mistakes that we did? We did not have a proper design, we just went there and with whatever design we had in mind, we just did not record it from high-level point of view and it had become a little bit of problem when we did the support."*

I initially coded the above data extract as *'support in agile.'* However, to sustain "internal homogeneity and external heterogeneity" (Braun & Clarke, 2006, p. 91) of the themes, I later recoded the above data extract under the *'evaluation'* theme. The *'evaluation'* theme was both frequent and significant compared to the *'support in agile'* theme, which I ultimately dropped from the list of candidate themes.

<u>Diverse data extracts</u> – Another major refinement of themes occurred when data extracts represented central points that were too diverse. This diversity mitigated against finding a coherent higher-level meaning for these themes. For instance, in the *'Agile versus Waterfall'* candidate theme the frequency of the data extracts that were signaling comparisons between agile and waterfall initially indicated the existence of a pattern. As more data extracts were coded under this theme, however, no convergence occurred in the underlying meaning that these data extracts were conveying. In other words, although the candidate theme reflected a comparison between agile and waterfall, the diversity of the focus in these comparisons meant no conclusive higher-level meaning could be extracted to explain the results.

Many of these data extracts had a secondary focus that also could have been used for analysis. Nonetheless, initially all data extracts were coded based on their main focus, which was the comparison between agile and waterfall. When this theme ('Agile versus Waterfall') did not lead to any higher-level, consistent meaning, nor did it link to other proposed themes, I recoded its associated data extracts based on the secondary focus in the data extract.

The revision of coding in diverse data extracts was done primarily to maintain the internal homogeneity of the theme. The following statement by Practitioner 3 is an instance of this code refinement:

> *"The analogy I was going to use is building the house. If you are going to build a house, you don't go to see an architect or a builder and say, 'I want a house!' And then you would not say, all right! Let's do a bedroom and when you do a bedroom, you would not say oh, let's do another bedroom. That was really good! So, let us do another bedroom. Oh, now let's do the lounge room… where do you want it?... next to the bedroom! Ok! And then after that, you stop half way through and go, umm, is that the master bedroom? Yeah…. Then, an en-suite would be nice, wouldn't it? And then you start to … to build an en-suite!! That is an extreme example but to me, that is the bad side of agile."*

Initially, I coded the above data extract as *'Agile versus Waterfall.'* During the theme-revision stage, however, I realised that the above data extract is addressing the role of *'high-level modeling.'* Based on the principle of the hermeneutic cycle in interpretive data analysis, I identified *'high-level modeling'* as a secondary focus for this data extract—through recursive refinement of the codes and the themes when considering the meaning of the entire dataset. As a result, this data extract was ultimately recoded under the *'high-level modeling'* theme.

4.1.3 Defining and Naming Themes - Point of Saturation

As the revision of themes progressed, an interesting and important pattern began to emerge. I realised that more revision and refinement of the themes were not leading to the identification of any new codes or themes. This outcome suggested that the structure of the proposed thematic map was a suitable fit for the entire dataset. In addition, the overarching themes began to make sense not just in relation to their immediate data extracts but also with respect to the other identified themes (in a way that the set of identified themes were narrating a cohesive story of the entire dataset). Also, I conducted several interviews (specifically, the last three interviews with Practitioners 4, 6, and 7) quite some time after the previous interviews. These later interviews overlapped the data analysis phase and did not introduce any change to the overall structure of the thematic map. As a consequence, I concluded that I had reached a point of saturation.

As I was analysing the data, I was writing memos about the underlying story that the data was narrating. These memos, particularly those written towards the end of data analysis, through recursive revisions of themes, inspired the naming of the final themes.

For instance, in a memo I initially named '*maturing industry,'* I wrote:

> *"It seems that agile practitioners perceive change in an evolutionary sense. They do not see change as a barrier to their success. Instead, it seems that they try to capture change as simply and as quickly as possible, as a feedback*

> *loop into their practice. It also seems that in an overarching interpretation of IT industry, they perceive the industry as a whole, and as a living organism that is maturing as it technologically evolves."*

This memo, which was largely influenced by the underlying story in the data, became part of the reason that I named the overarching theme *'maturing Industry.'*

Ultimately, the thematic analysis of the semi-structured interviews resulted in my identifying five overarching themes that underpin modeling practices in an agile context. I named these themes *Cognitive Engagement, Fragmentation, Volatility, Living Organism,* and *Maturing Industry*. In the following subsections, I explain each theme and its associated subthemes as contextual elements of agile-modeling practice and a lens to reflect on the interviews to obtain a cohesive interpretation of the data.

4.2 Real-life Practice of Modeling in Agile – Context and Methods

In the following subsections, I describe five major themes that characterise the context and methods of agile modeling practice.

4.2.1 Cognitive Engagement

A key notion in the data extracts was practitioners' emphasis on the importance of *cognitively engaging* all stakeholders in system development projects so that a deep understanding of the task at hand could be attained. This key notion formed the overarching theme of *Cognitive Engagement*, representing five subthemes of *common understanding*, *coding as sense making, communication, ownership,* and *translation.*

The choice of the term *'cognitive engagement'* was based upon the work of Milton, Rajapakse, and Weber (2012) in an experimental study on the quality of conceptual models. In their work, they investigate the concept of cognitive engagement in relation to quality evaluation methods. They hypothesised that stakeholders would be engaged more deeply with the semantics of the domain represented by a conceptual model if they used a well-defined evaluation method. Furthermore, if the quality-evaluation method invoked challenging tasks for the stakeholders, they hypothesised that the cognitive engagement of stakeholders in eliciting a full understanding of the domain semantics would increase (Milton et al., 2012)

Milton et al. (2012) adopted an ontological perspective of conceptual modeling as the underpinning assumption of their work. Therefore, their research seems to be more aligned with the ideas of waterfall methodologies. In my research, however, I undertake a hermeneutics approach, similar to Sarker and Lee (2006), to interpret the notion of *Cognitive Engagement* in the context of agile methods. To do so, I first report on the *findings* for each of the underlying subthemes of *Cognitive Engagement*.

4.2.1.1 Common Understanding

Almost every interviewee indicated that conceptual models are the reference points that document an achieved level of *common understanding* about systems domains. In describing the context in which such common understanding is achieved, the interviewees emphasised two factors. One is the extent of differences among stakeholders' perspectives, and the other is the extent of differences among various system applications. They also perceived conceptual modeling as a process of *cognitively engaging* stakeholders both with the domain and with other stakeholders, so that differences are reconciled and a single, consistent model is produced.

Practitioner 7 marked the importance of reaching *common understanding* as follows:

> *"… Because, everybody integrates with the business at different points, they have different perspectives and there is a general understanding about what the business is doing in an industry, different entity types, different market sectors and there is different functional groups within the business as to whether you got accounts, whether you got logistics, whether you got manufacturing, whether you got the retail side of things. So, you have got different components within the business and then everybody has an understanding, a conceptual understanding as to what the functions of these groups are and basic understanding of the business processes involved in this. Fundamentally, there are inconsistencies within organisations as to how the business works and fundamentally, there are inconsistencies with the way information is handled within organisations and sometimes these misalignments are insignificant; person A calls that a widget and person B calls it a plugin. It is the same thing, but there are just different phrases for it. However, you can get quite significant misunderstandings, which have a profound impact on the performance of the business. A lot of the work that we do when we try to interchange information between the business is when you are coming up with these fundamental differences between these systems."*

The interviewees indicated that stakeholders who belong to different levels in an organisational hierarchy often have different perspectives about a domain and a possible solution model. They possess different skills, come from different cultural backgrounds, and have different life experiences. Furthermore, stakeholders apply their distinct viewpoints to different application fields. Each of these differentiating elements raises a barrier to reaching a *common understanding* as a basis for conceptual modeling.

As a result, interviewees indicated that in practice they often created multiple conceptual models instead of a single, comprehensive model. A number of practitioners who had experience in larger organisations at the enterprise level reported that multiple conceptual models expressed in different grammars and with varying levels of detail were used. Each model provided a *partial* representation of the domain. In this regard, Practitioner 1 stated:

> *"… When I came on board, they had multiple models in varying languages and in the level of details and they wanted 'one' model that could be their reference model. In technical terms, you might refer to that as a canonical model… um, in layman's term, it is the reference model. It is the model you referred to, it is the agreed standard."*

Nonetheless, in spite of variations in the ways stakeholders perceive the world and the existence of distinct application fields, many interviewees argued in support of the need to prepare conceptual models that constitute reference models representing a *single source of truth* for the entire system. They contended that conceptual models should be *canonical information models* representing the system domain in a single reference model. Although multiple conceptual models may initially represent different understanding of a domain, they insisted that conceptual modeling practice is about integrating these different perspectives of the domain and different application fields.

In conceptual modeling, the interviewees argued all stakeholders should be *cognitively engaged* in a practice that seeks to logically integrate multiple representations of a domain into a consistent Common Information Model (otherwise known as a Reference Model or a Canonical Model). In this sense, conceptual modeling is a process in which stakeholders from different spheres of influence cognitively engage in a diverse setting of multiple understandings of a domain. The objective is to reconcile views and achieve *common understanding*. In this regard, Practitioners 7 and 6 said:

> *"Fundamentally, a model from my perspective is a […] reference point to get an understanding about something and typically, when you are dealing with business and development, you find different levels of people within an organisation and you have different levels of people within the project team and development team and each have their own perspectives on reality and their own perspectives of what the model of a solution needs to be. And, doing information modeling is all about trying to represent a common understanding so that when you refer to something, everybody knows what it is and everybody is aware of where it fits in the business, everybody knows where it fits from a process point of view, and everybody is aware of what it means from a development point of view and support point of view for the end product that we build."*

> *"… Conceptual model[ing] is really all about having consensus of business and stakeholders. From logical point of view, we do have evaluation in the sense that it ticks certain boxes and answer all questions before we release to the physical model. But conceptual models are really about having consensus…"*

4.2.1.2 Coding as Sense Making

One of the interesting notions raised by practitioners who were more directly involved with pure agile projects from start to finish was the notion of coding as a probing method to gain more information about the system. In *coding as sense making*, the purpose of *cognitively engaging* with the system in the form of coding is not initially to provide a system deliverable but to explore possibilities for better understanding of the domain. Practitioner 3 indicated:

> *"... When there is certain lack of clarity and when you do not know exactly what you want, but you know enough to know some details, there is absolutely no harm in saying right! Let us just do something to get some sort of understanding of what the actual nature of what we are trying to solve is! The nature of the solution that we are trying to deliver and then, that helps you get more clarity around what the end state would look like."*

From this perspective, the practice of coding is identical to the practice of modeling as an instrument, enabling stakeholders to cognitively engage with the system and get more information about it. Practitioner 2 described conceptual models in agile as follows:

> *"... Conceptual models because they are a basis to get more information from the business are not design artefacts; they are not implementable."*

In this light, *coding as sense making*, similar to conceptual modeling, provides some structure that enables specific enquiries about a domain to be made. The practice of coding as sense making engages stakeholders cognitively with the domain through providing a technical base in the form of codes. Codes enable better domain understanding through probing and interacting with stakeholders. Practitioner 5 elaborated in this regard:

> *"…We are trying to code in such a way that [we] can react. […] I want to say to the end users, here is the data. Use it however you like. You can figure out what you want to do with it and structure it far better than I can with all theory sessions and business analysts and white board stuff you want to do. You just go and use the data – go and use it! You figure out what value might be in it, because you are an actual business driver who can find value in it. The last thing I want to do is to get the IT and information systems in your way. We provide you the technology that allows you to do that initial exploration. From whatever data you want to go and look at, you see whether there is value in it and whether there is persistency in it rather than tying up expensive IT resources and chasing windmills."*

4.2.1.3 Communication

One of the other notions frequently addressed in relation to cognitively engaging stakeholders in agile modeling practices was the notion of *communication*. Almost all interviewees agreed that *communication* is at the heart of the practice of modeling in agile. It has the purpose of validating perceptions and building a solution model. Practitioner 2 explained:

> *"Conceptual models are first cut of our understanding of the real world or the description of the business and the main purpose is pure communication and validation of those concepts."*

Practitioner 8 depicted engagement of stakeholders as a communicative act to elicit system requirements. He commented:

> *"… And then, separately, we do a stakeholder request and what we do there is that we say, hey, what is wrong with your analytical platform right now and how would you fix it, if you were in IT? So, now, we have three different parties, sponsors and two different stakeholders and we hear all these complaints about why the company can't do what it needs to do and what would be a better world, and then that is where the project architects goes down and that is where we get this vision document and a vision document is a very short document with two lists and three diagrams, but it basically saying to all those people whom we talked to in the business here what are the problems we hear and here is the sketch of a solution, and that has a data component in there and to show you next is the component about the visual document and you can see where the data comes in. So, the point is that we just don't sit down with the data. We had interviewed with the business first and it is a very pointed interview; firstly, how you are going to make money with this project – so, give us a value proposition. Number two, we talk next level down the organisation and we say, you know, what is wrong with your current system and how would you fix it and now, with that, we start doing data and with data, this is the vision document. And, then we do the problem statement. If we understand the problem, that the company smoothly suffering from and we understand how it impacts the company, then we propose that this has got the solution for the problem and here how it is look like and here is an increase or decrease our measures."*

Like *common understanding*, practitioners referred to the complexities of *communication* in a setting that is extensively diverse as a result of stakeholders' different backgrounds and perspectives. This diversity results in a multiplicity of model representations. Practitioner 7 commented:

> *"… So, again we start a project saying ok, this is what the business wants but then you have people at the top of certain focus, of certain vision. So, you have got to skew then between where the top end and the senior managers want to go, versus where the guys of the core base who are actually doing the work because, there is a time lag between the two. So, again this conceptual model type of thing does shift as well because you know, you can talk to the board of directors and the senior management to say this is the vision we want to fulfil and that could be two years ahead while the guys who are actually in the departments are entering the numbers where the staff of the warehouse are getting numbers […] So, again trying to represent one conceptual model that is representing different perspectives of different timelines are quite difficult."*

Furthermore, to facilitate *communication* among stakeholders, practitioners pointed out the importance of minimalistic representations in the form of high-level modeling. Practitioner 2 commented:

> *"So, in a conceptual model, we are interested in validating something specific with a user in a session and therefore, we do not discuss what the details of the full design [are], and how it is going to be implemented. But, [modeling] is performed in a very simplified way because the purpose is just to facilitate communication between a business user and a technical person. […] I would say [the main difference between agile and waterfall conceptual modeling] is*

the level of details. So, my purpose is communication. Therefore, I intentionally hide anything that may be confusing and make the communication unclear. So, if I don't need to go to a session to validate attributes of a product because it is too low level and too distracting, I won't put it there… but if I have to discuss and validate attributes to understand a problem and it is critical to understanding, I probably put it there as part of my conceptual schema because my whole purpose is communication. But, usually the high level is enough to achieve the validation from the business."

As indicated in this data extract, the importance of high-level representation in agile methodologies is so significant that they are considered to be the point of distinction between agile and waterfall modeling. While conceptual modeling in waterfall methodologies is concerned with a detailed representation of the domain semantics, agile practitioners prefer minimalistic, high-level representations of domains. They argue that high-level modeling facilitates communication and better engages stakeholders with the domain. Practitioner 7 commented:

"So, I think an approach [to modeling] needs to be more creative, more communication-focused and less logic-technology focused. We have got a very "skewed" approach to [modeling in waterfall methodologies] and this is why we got this myriad of formats of modeling, myriad styles of modeling. This is how this discipline looks like. But what is the model there for? It is there to communicate and you might have seven models, which is to have seven conversations but everybody needs to understand that."

4.2.1.4 Ownership

Agile practitioners found that validation of models that are presented to stakeholders who are not actively engaged in the process of modeling is problematic. They argued business stakeholders find it difficult to engage cognitively with representations of the domain semantics if they had not been engaged with the process of eliciting such representations. Practitioner 1 said:

> *"… Sometimes people are asked to review a model and they do not know what they are looking at."*

To address this issue, in agile methodologies, wherever *visible* modeling practices are taking place, practitioners adopt a strategy of cognitively engaging every relevant stakeholder in the process of modeling. Practitioner 1 described this process as follows:

> *"So, I can communicate on a single page and towards the end of the week, I created a schematic which is a diagram, took me a few minutes and I shared it with people and now we have the start of the whole thing and this is perhaps where I am a bit radical and I am not sure how other people would approach this. But what I did was that I actually ran a two-day course for business people and technical people and I got them together in the same room. Day 1 was basically the context of what modeling was and […] on day 2, I divided them into groups and each group deliberately had a mixture of technical and business people and I assigned each group, one of these domains […] to work together and consolidate all of these models of the domains into one model for the enterprise."*

As a result, stakeholders have an improved sense of *ownership* for the generated modeling scripts. This sense of ownership is both a reflection of stakeholders' deep engagement with the process of domain understanding as well as its outcome. Practitioner 1 further elaborated on the importance of stakeholders' ownership:

> *"So, I got feedback on two particular areas when they said you are 'wrong' politely [chuckles] and even among themselves, there were a bit of debate. But they reach an agreement that was different to my position and I changed my high-level contextual model and the important thing is that they now 'own' the model. It was 'their' model [chuckles], which is a fantastic outcome actually."*

4.2.1.5 Translation

Similar to the *common understanding* theme, where the importance of eliciting logically integrated representations of domains was underlined, the *translation* theme highlighted the importance of eliciting a shared terminology among stakeholders. Practitioner 2 described the absence of a shared language among stakeholders as follows:

> *"… [Validation of conceptual models] is perhaps a complex part of the process. Why? Because it would be ideal if everybody would be able to read these [scripts] in the same way, and interpret it similarly. But because not everyone here has same language… in IT analysis and design, we do not have a "lingua franca" with the business users. Some business has better skills but in general, they don't. They do not speak what IT language is and in most cases, the approach is: 'I' know that for instance, this is one to many and 'I' know that business say 'is it true that a service can be related to multiple provision products?' So, 'I' have to read it to the business and the business*

says yes or no and I have to correct it or confirm it … but it is always through interpretation because unfortunately the business cannot make sense usually directly of these conceptual models."

Practitioner 5 addressed the importance of *translation* as follows:

"My background is accounting and then I moved to IT. I have started off as an accountant and then I moved to IT. So, I had that translation between the business background and the technology background. So, in the past 20 years of my life, I had been working effectively as a translator in that sort of space [chuckles]."

However, clarification of different terminologies does not occur only between technical and business stakeholders. Practitioner 5 described how business terminologies too were refined during the practice of modeling as stakeholders cognitively engaged with the domain semantics:

"… Some people say that is the price that customers pay; so, that is how much they paid and it is the total amount. Some others say, it is the value customer pays after deducting GST, after we deducted government fees like terrorism levy and those sort of stuff. When we talk premium, again in the context that we include this and do not include that, we have the conversation about written premium and so, that variations, when you are trying to do information systems, the definition association and the official definition which is actually a localised variation of it is quite impactful if you will."

Practitioners indicated that the *translation* between distinct terminologies is an important aspect in the evolution of modeling practices in agile methodologies. They argued that Common Information Models are the agile counterparts of conventional conceptual models. Practitioner 1 articulated the evolution of conceptual models as follows:

> *"Years ago, only one design would be done; just in a physical layer. And gradually, it was argued whether it would be good to design first and defer the physical implementation until the logical model is sorted out. So historically, most people did physical and then the more enlightened we became, we did the logical first and then some developers still say who needs the logical in waterfall and then when people started business analysis and all the rest, they said this is not implementable in core conceptual or whatever. Now, with Common Information Model (CIM) we have a canonical model, which is the hub-and-spoke… I don't have to express all these business domains in terms of details of implementation, I just express it in terms of CIM and that access the translator and the example I give you is if you attend the UN, you might speak Portuguese and I might speak Swahili or whatever… if our first language are different, as long as everybody in the room has a common language which is English or French also might be one of the official common languages, if a bunch of people know a common language in addition to their native tongue, they can talk. Instead of learning all the languages in the world, if we all learn one common language, we can communicate and that is the cause behind CIM [in providing us with such a common language]."*

An added benefit of a Canonical Information Model is that it provides a reference point for the meaning of terminologies used in the context of the business. Practitioner 1 added:

> *"Another use for CIM is semantics for business vocabularies and business rules. So, if I have a business rule, since any business rule is going to use nouns, and the CIM can be also used to shape the definition for those names, for the vocabularies in the business."*

4.2.2 Fragmentation

As discussed earlier under the overarching theme of *Cognitive Engagement*, logical integration of different modeling scripts in a Common Information Model is an objective in agile modeling practices. To meet this objective, practitioners stated they cognitively engage stakeholders with domain semantics through provision of *common understanding,* an increased sense of *ownership*, *coding as sense making*, *communication*, and *translation.* These actions mitigate the negative effects of extensive separation among stakeholders' perspectives and application fields in the practice of system development.

Remote worldviews of stakeholders and distinctive fields of application are not the only factors contributing to a sense of separation in the context of modeling practices. Practitioners also referred to other factors that contribute to *fragmentation* both in the modeling scripts and modeling practices. Three subthemes were apparent in their narratives: *technological fragmentation* as a result of disconnects in technological infrastructure; *theoretical fragmentation* as a result of disconnects between theory and practice; and *solution semantic fragmentation* as a result of disconnects between representing a solution semantic model and the set of requirements. I discuss each of these subthemes below.

4.2.2.1 Technological Fragmentation

Practitioners who had experience in large organisations with a history of established information systems reported that technological fragmentation due to the existence of legacy systems hindered the use and maintenance of conceptual models. In some cases, disconnects between current and past technologies had resulted in conceptual modeling practices becoming obsolete. Practitioner 4 stated:

> *"... um, there are places ... particularly very large organisations in my experience, the very large federal departments, they are still using very large databases that are developed decades ago on technologies that are pre-dated with a lot of the modeling tools and so, all the modeling are done in very odd tools like Oracle case tool, which is really no longer used and so those models tend to die and maintenance of the databases are done very manually."*

4.2.2.2 Theoretical Fragmentation

Those practitioners who had lengthy experience across various industry sectors confirmed a general disconnect between established theoretical views on conceptual modeling and the practice of conceptual modeling. For instance, unless project owners specifically requested conceptual models as project deliverable, practitioners indicated that modeling practices often are not done based on theoretical considerations. Practitioner 4 indicated:

> *"... So, my overriding experience has been that in the domains that I have worked in, data modeling, conceptual data modeling has not done upfront as a way of understanding or explaining the domain."*

The uneven distribution of expertise in projects was one of the reasons that were given for the described theoretical fragmentation. Practitioners reported that theoretical considerations are only followed to the extent that project team members possess the relevant skillsets. In other words, practitioners reported that the exercise of modeling practices is not always guided by theoretical implications. Rather, the type of skills available in each project team largely marks the practice of modeling. Practitioner 2 commented:

> *"… Yes, it would be at least the expectation that the IT [professionals] speak the same language. Unfortunately not! And the reason is, there are many people specialising in different areas. The fields are so wide. Many developers fascinated with Java, object oriented, etc. or this or that language and they all have to work with these designs but unfortunately, they do not have the same level of expertise. So, basically modeling is not a core skill among IT practitioners and I think that is a pity. Because everybody in IT, if they are building a system, they should understand modeling."*

Similarly, Practitioner 4 commented:

> *"… So, that standard data modeling division between conceptual, logical and physical in all of the works I have experienced is not done. The reasons for that are, firstly, data professionals are always the minority on any development project. So, if we use this client survey developers model that [was] common in the 90s, 2000, you tended to have all C Sharp developers or may be some other number of application developers of any other tools that were available there, DB developers… they formed the majority and then the next layers are those who do tests… in the application development, the test is*

very important and then probably the next one are the BA's, the business analysts and then the data people are actually coming in the forth or the fifth in terms of the number of people on any development projects and therefore, simply by value of having small numbers they tend to have much less influence. So, a lot of that is that if we are saying we should model the problem this way, it is not going to get in."

However, the theoretical fragmentation in the practice of modeling is not just influenced by the dominance of people-adopted skillsets in the projects. Furthermore, often a dominant trend in the whole industry of Information Technology influence the approaches practitioners adopt to address the problem space. For instance, Practitioner 4 explained:

"… Then of course in the year 2000 by the explosion of data, Java developers came as the main developers on the projects and of course they had a huge impact on modeling, because they used UML. And then, doing model-driven development with UML, generated certain core class artefacts in Java. But again, the data modellers were very much a small group who tended to have to follow. So, again if we wanted to model the problem space in a particular way, then sure we could, but that was not what the project were doing. So, with the same affect, these days of course, it reflects on web developers. It's web service developers… they are now forming the practice and when it comes to modeling your information, architects have higher visibility and inform the practice rather than data modellers. So that is one reason I haven't seen data modeling done the way it should have been done theoretically."

4.2.2.3 Solution Semantic Fragmentation

Practitioners generally agreed that identification of system requirements is a major design step when developing information systems. They maintained that traditionally the technical and functional as well as non-functional specifications, processes, and data flows were developed based on a specified set of requirements. However, they stated that identifying system requirements is different from building a solution semantics model.

When building a solution semantic model, which is often neglected in the process of modeling, system requirements are addressed in relation to business value generation chain. In this regard, Practitioner 4 commented:

> *"… Very often, it [solution semantic model] is just dropped off which means a mountain of problems coming in the delivery of solutions. Because I think we are agreeing here that there is, apart from the requirement, which is a strict statement of a very itemised and low-granularity piece of functionality, there is actually solution semantics, which ties that altogether and that very often says what to do and that very often is not done and is not captured in my experience. I have worked on one project, which they really did try to do that and they did a lot of functional modeling in conjunction with semantic modeling and that is the only project that I have personally seen they have done that, very rare. All the others have worked off the requirements and even then, the requirements were not managed very well. So, when it comes to solution semantics which is what comes from a data perspective, that is what producing a dataset or what our data attributes are, is meant to be this and that should line up and consumable with that bit just over there and that is*

very often not captured and not validated against… in […] agile, as I understand, we have got a set of requirements A, B, C, D. What we can do with agile, we can go to the business and say A, C, and D, no problems. But we should do B quite later in time and the business says ok, because they are agile. So, in a sense, it puts even more emphasis on requirements away from a solution semantics model.

4.2.3 Volatility

Volatility was another notion that clearly stood out as an overarching theme in the analysis of the interviews. Volatility in lexicon means *a trait of unpredictable change*. Accordingly, data extracts that reflected the notion of *volatility* underlined the unpredictability of categories of change that can happen within the information systems domain. Based on this perception of domain volatilities, practitioners argued that endeavours to fully capture or represent the current state of a domain in a modeling practice should be critically assessed against the value these representations can potentially bring to the organisations. Practitioner 5 argued:

"… If the world was stable and businesses never change and people did not evolve, we could have got our models 100% right and all the notations that now exists would be able to do that… and that [would be] great… You could create a model, a representation that describes an organisation at any point of time, but why? It will take you a number of man-years or effort or whatever you call it to get there [...] and you can probably do it [model perfectly for a physical system], because it is relatively a finite set and it is expanding, but expanding in a measured way; you can probably do that and capture all the information for the structure that probably already exists and that is fine

> *because we are in a fairly rigid environment. But, I come from a business background and the whole point there is to bring value to the organisation; my job is to bring value to the business; they pay me to do that."*

Under the overarching theme of *volatility*, four subthemes exist: *lack of continuity of stakeholders, uncertainty due to human agency, politics,* and *lack of paradigm fit - need for the discordant view.* I discuss each of these subthemes below.

4.2.3.1 Lack of Continuity in Stakeholders

Many practitioners indicated that one source of volatility in information systems development is the unpredictable changes in stakeholders. As discussed earlier, the practice of modeling is heavily reliant on the cognitive engagement of stakeholders with the domain semantics. Thus, unpredictable changes in stakeholders could undermine the collective engagement of stakeholders in modeling.

To address this issue, and because elongated project terms and large project teams are more susceptible to such lack of continuity, agile practitioners argue for smaller teams and shorter delivery times in information systems projects—qualities that depict more agility in practice. Practitioner 2 argued:

> *"… It is complex. Normally in a project, one single person or single team is responsible for the modeling, not just the actual representation, but also the consistency of the design across the system. But, because a system development in average takes about 8 to 10 years, many different people come on board and are involved with enhancing and maintenance of the system and their approach to design and to modeling may vary and therefore, this becomes an issue…"*

Practitioner 3 depicted *lack of continuity of stakeholders* as follows:

> *"… The other thing that can happen is that when you have simply things like people who originally said this is what we need, and you have prepared the business case by them, may not be the same people that get involved in the detail of the problem definition and they may not necessarily be the same people who are involved in the requirements gathering process, and they may not necessarily be the same people who are dealing with the vendors and the providers and so…. as you are going along, there is a game called Chinese Whispers… I don't know if you have heard of that… but basically, you sit with a big group of people around a circle and you start saying something to the first person and by the time it goes all the way around when you get back to the start, you get something totally different."*

4.2.3.2 Uncertainty due to Human Agency

A number of practitioners emphasised the impact of *human agency* on information systems modeling. In other words, their concern was the impact of human actions and decisions that seem to be non-deterministic. They argued that humans are an important part of organisations and the non-deterministic nature of their agency impacts practice. Some practitioners insisted that businesses and organisations experience volatility as a result of the uncertainty that human agency imposes on practice. Practitioner 5 commented:

> *"… and human beings are changing and that has become the problem of the last 10 years in the scope of information management type of systems that they are in fact organisms rather than systems and they move in biological ways rather than logical or structured ways. So, you have to think of them in*

> *biological terms […] the nature of business and organisations is that we operate in human world. Humans are moving; what their focus on; what their environments is constantly changing, in all sorts of directions. In the world I work as an information manager or in the information delivery, what I work with today or what I look at today and what is important to me today is not what I was looked at, yesterday and in fact, what I look at this morning has changed from what I have looked at this afternoon… [In some organisations] that modeling is very scientific; they revisit their models every 3 to 6 months, because of the fact that they are monitoring those factors that change the model. So, they have to go back and revisit the model so they can identify changing factors.*

4.2.3.3 Politics

Some practitioners referred to conceptual models as *single source of truth about the system*. Some emphasised the role of *politics*, however, as a barrier to achieving truth. Practitioners' reference to politics concerned deliberate efforts to conceal the truth because of hierarchies and power struggles in organisations. For this reason, although politics can also be understood as a manifestation of human agency in organisations, I distinguish it as a separate subtheme. In this regard, while human agency is broadly concerned with the impact of humans on the dynamics of practice, politics specifically addresses the issues related to the dynamics of power in organisations. Practitioner 4 described some aspects of politics as follows:

> *"… and then the third one that has been written about for decades is the persistence of the information silos and the degree that information silos are tied to the actual politics. That is a major frustration for developers… and the*

> *larger the organisation, the more their information assets are divided in silos that are attached to organisational divisions with political alignments."*

Practitioner 3 indicated:

> *"… Yeah, it is a little bit of everything… no one ever has the 'perfect information' to use an economics term. So, you see… it can be anything from people… well… when you are working in large organisations or larger projects, politics always comes into it, which is unfortunate and it is always from higher level vendors and clients having budding heads to… within the client different people having circles, you know, things like that… um…"*

4.2.3.4 Lack of Paradigm Fit - Need for the Discordant View

A number of practitioners indicated that a source of volatility and therefore failure in information systems development is the impact of technology in projecting an overall picture of a domain that is based only on uniform, conforming views. By dismissing discordant views in the process of systems development, they argued that important information about a domain might be lost. To preserve meaning about a domain, they argued that system design and modeling practices should be guided by two distinctive paradigms: a conforming paradigm and a discordant paradigm. Practitioner 5 commented:

> *"… and, that is part of the visionary stuff that I am doing and it is that I don't want to model this state, I don't want to worry about how it moulds... I want two paradigms around! I want to say to the end users, here is the data. Use it however you like. You can figure out what you want to do with it and structure it far better than I can, with all theory sessions and business analysts and*

white board stuff! You just go and use the data – go and use it! You figure out what value might be in it, because you are an actual business driver who can find value in it…The last thing I want to do is to get the IT and information systems on your way! We provide you with the technology that allows you to do that initial exploration. From whatever data you want to go and look at, you see whether there is value in it and whether there is persistency in it rather than tying up expensive IT resources and chasing windmills. IT comes in after the event. After you find that insight… to that allowing people to get to the insight […] those people [people with discordant views] are worth listening to… apart from that they usually don't listen to what you are saying [chuckles]… if they do listen, those people are worthy to listen to cause most of the others agree with what you are saying. But those are worthy because that is where insight unusually comes from…"

4.2.4 Living Organisms

Contrary to the three previous overarching themes, which I named based on the meaning of their constituent subthemes, the next two overarching themes—*viz*. *Living Organisms* and *Maturing Industry*—are '*in vivo*' codes (Urquhart, 2013). *In vivo* codes are codes that practitioners suggest. They incorporate practitioners' views on the interpretation of the data.

The aim of creating an *in vivo* code is to ensure that concepts stay as close as possible to practitioners' own words. Urquhart (2013) argues that *in vivo* codes are among the most significant codes for researchers because they capture a key element of what is being described through data. In a description of how conceptual models are perceived as '*living organisms'* in practice, Practitioner 7 said:

> *"… So, a model, a conceptual model should be a living organ in itself, a living organism. It is something that needs to be constantly reviewed and [inaudible] and if you don't use it, you lose it. It is the same thing with the brain. If you don't use this particular relationship, or attributes or things, they get depreciated and become less important."*

Practitioners who use agile methodologies identified attributes of desired conceptual models largely based on their perception of the elements that characterise the context of practice. One of the main characteristics they identified is *complexity.* Consequently, practitioners argued that *adaptability* and *distinctive identity* are two main attributes of desired conceptual models in agile methodologies. I discuss the contextual element of *complexity* and the resulting features of *adaptability* and *distinct identity* as subthemes of the *in vivo* code of *Living Organisms.*

4.2.4.1 Complexity

Almost all practitioners, either implicitly or explicitly, referred to the notion of *complexity* as one of the most evident contextual elements in the practice of modeling. Implicitly, all practitioners indicated that *complexity* is an intrinsic characteristic of practice. Comments on complexity were so pervasive that all the themes discussed above could have been interpreted as subthemes of *Complexity*. In this regard, *Cognitive Engagement* revealed different strategies to better manage domain *complexity*. Similarly, *Fragmentation* and *Volatility* highlighted different causes of *complexity* in systems development.

Nonetheless, data extracts reflecting the notion of *complexity* as a theme were not restricted to the subthemes identified above. Indeed, practitioners explicitly discussed *complexity* in the context of modeling. In a data extract explaining the nature of *complexity* in practice, Practitioner 7 described the role of information system experts as follows:

> *"... The role of an information architect is understanding the business, understanding the vision and the road map that the business needs to go down, understanding the history and the legacy of the business as to where it has come from, what the drivers are, what the issues are, what the inconsistencies are in regards to all the information interchange. Fundamentally understanding the way the technology being used, the applications; understanding the detail of where the information coming through, the technology and the people involved in that technology and then coming up with a vision, an understanding of how all of these hangs together. That is why it is a hard job!"*

In such a context, practitioners argued the desired goal of conceptual modeling must be to reduce *complexity*. Practitioner 2 commented:

> *'... In complex projects, we say that if [the model] looks too complex, normally it is wrong. Normally, models have the quality that they need to be simple and elegant solution and encapsulate in a simple way what you want.'*

For instance, the notion of *iteration* in agile methodologies was discussed mainly in relation to the need to break down the *complexity* of the system development tasks as well as to break down the complex semantics of the domains. Practitioner 2 stated:

> *"Agile is in fact good in allowing and recognising the iterative nature of building a system. I can come up with this today; and do two-three validation sessions with business and if it is 80% correct, I can start development. If later*

> *on, there is a new requirement from the business to change my assumption, agile just accepts it and changes the model and then we manage the development to capture that new requirement. That is agile and that is more natural than saying that, ok... you have two months to finalise 100% of correct conceptual model, your design and you commit to sign off that, and the business signs off and if something changes, bad luck! The system is delivered regardless. That is the difference between the cascade [waterfall] approach that is not natural in terms of not recognising and capturing the changes..."*

In dealing with complexity and domain change, practitioners also referred to *high-level modeling* as an approach in agile methodologies. *High-level modeling*, similar to *lean modeling* (discussed earlier), focuses on representing phenomena in the early stages of modeling without the need to identify fine details that relate to attributes, cardinalities, etc. This approach to modeling motivated recognition of *Model Granularity* as a notion that highlights a distinction between traditional conceptual modeling scripts and the high-level models used in agile practices. In agile methods, practitioners argued that conceptual models often represent coarse-grained phenomena. Consequently, a high-level dialogue among stakeholders without the need to capture and validate detailed phenomena becomes possible. The design and verification of fine-grained models that is taking place in logical and physical models are largely informed by technical details in agile and not the proposed principles of traditional conceptual modeling.

In essence, *Model Granularity* in agile methodologies is a major point of departure from the traditional approach to formalising conceptual modeling scripts. In traditional methodologies, the emphasis is on representing domain semantics in detail.

In agile conceptual modeling, however, scripts are only capturing the main subject matter. In this regard, *Model Granularity* is a distinct notion that stands out in the analysis of data. Practitioners 6, 3, and 8 commented:

> *"… It is different with conceptual modeling in the sense that if you do not know the big picture, you do not know what puzzle you have in the first place, right? […] I have a bigger picture… I don't really concentrate on the bigger picture but I do actually have the context… that is the conceptual figure; but then if I have to go deep down for the particular iteration, then I use that piece of work to conceptualise next to logical…."*

> *"But to me, you still need that first high-level understanding of what it is you are modeling cause when you think about it, it is about the information that you try to capture and different things that you are trying to deal with."*

> *"And in that point, if you can get them to validate this [conceptual model], then it is worth to think about the attributes. But, one thing that we notice is that data modellers jump into the detail way too soon before they know what the problem is. Before they even know the high-level solution, what they are already doing is to get business people try to validate all the attributes in the table whereas it is meaningless for the people until they understand how the entities are going to solve their problem."*

4.2.4.2 Adaptability

Consistent with the literature on agile development, all practitioners emphasised the role of *change* in shaping modeling practices. Change, as one of the most fundamental concepts in understanding the context of agile methodologies, has different dimensions.

I discussed some dimensions of change under the theme of *volatility,* where unpredictability of change was emphasised. However, *volatility* is only one manifestation of change in system development practice. Not every form of change is unpredictable. As with *complexity*, practitioners perceive change as an intrinsic characteristic of domain semantics. They argued that conceptual modeling methods and scripts must be capable of high levels of *adaptability.* Otherwise, information systems that are developed on the basis of inflexible models will be short-lived.

For the most part, practitioners rejected the notion that *faithful representations* are a prerequisite for successful information systems. They argued that faithful representations of domains are impossible to achieve. Moreover, they argued that domains are changing constantly. As a result, the resources needed to achieve faithful representations cannot be justified. They insisted that *business value* considerations should inform modeling practices. Practitioner 5 commented:

> *"... Structures change over time and can be reconfigured and by configuration... in that sort of environment [highly changeable], if you are receiving information systems that are hard coded on that structure, the minute they change, you break! [...] The insight I have come through in the past years, because I have worked with systems for a very long time and I became very dispirited of how many systems I worked on that have failed and when I say failed, I mean systems that we technically delivered, but wouldn't address the requirements or the business people would not use them, or all that sort of stuff and it was not about meeting my requirements today, it was*

about meeting my requirements tomorrow, it was always I have got a different view, you have taken too long […] So, me as a practitioner in that space, am I going to get it a 100% accurate for yesterday? No! I want to get it as significantly right for today and as possibly I can and try to position it where I think it might go tomorrow.”

Practitioners 2 and 8 described how change in systems necessitates *adaptability* in practices:

“Because, we are representing the reality and representations are varied and it happens over time and not only those assumptions change over time but we are unable to capture all assumptions correctly, so at some point you have to say that this is my final design that I want to implement. Later on, of course, if some assumptions change and those assumptions may make your maintenance more complex or less… in fact, that is where agile is good...”

“There are always places that we have to do it [faithful representations] because there are complicated business rules but that is like five to ten precent of the system and to creep the entire system description at the same level of detail is a killer. What you need is a flexible approach to deal with the stuff that is predictable at a high level and then selectively attack the stuff that are complex with detail.”

Practitioners were more focused on how to achieve *adaptability* in the approaches they adopted toward conceptual modeling rather than achieving faithful representations of domains. Practitioners 8 and 7 discussed how conceptual models are *adaptable* to change if their representational structures are simple and flexible:

"The intersection between what you are doing [research on conceptual modeling] and what we are doing [real-life practice] is this whole notion of incrementalism. Here is what the world really needs. The world does not need another theoretical approach to get the ontology right. What the world needs is, if you build all these entities and give it to your business and they use it and then they come to you and say, hey you know what, I add this, this is existing data and these are the relationships; Suddenly, there is this big impact on the existing data structure and all the relationships. So, the relationship between our world and your world is what the theory is to take an existing table and add a new relationship and how can you codify that into certain rules that machine could do that when the developers just draw this arrow to this new entity, so the machine knows how to adapt the structures and the existing data […] This is a place that some ontology can do. You have to come up with an incremental ontology to provide the rules to go from one ontology to the next. That is very valuable work and that is where we think we intersect."

"So, you need to design your model in such a way that […] you just load everything. So, it just gives you that ability to be more flexible: boxes, lines; boxes, lines… a lot of guys get off on these, oh no! [...] We need to have this 3NF; we need this semantic; we need that orthogonal model with that component. It is the question of why you are making it so complicated? They have got XYZ taxonomy here, that industrial model, blah, blah, blah. It is great, but there is a lot of stuff in there that is just confusing; it is a lot of noise. Therefore, just cut the chase and go back to the basic of what you are trying to do. That is my practical approach anyway."

Besides emphasising the need for simple modeling structures that allow model *adaptability* to domain changes, practitioners argued that conceptual models needed to have *adaptability* in relation to different project types and industry sectors. For instance, Practitioner 7 described how in some contexts where requirements elicitation is not the starting point for projects, conceptual modeling may be overtaken by purely technological approaches if the modeling methods cannot be adapted to the specific context of practice:

> *"Sometimes, you get engaged in a project that you never even get to hear about or see the business users; if you have a data migration for example. So, it is a technology-based project and it is a technology-focused project. The term business never comes to it. You talk about system A, system B, system C and this is a legacy system and when you need to [inaudible], you have got to get this system coming in, this is how they do it, this is the form that gets information from here to there. Off you go! And, a lot of people approach it from a technology solution [point of view]. They go, right! We are going to get square peg, round hole, square peg, round hole. So, we are going to get the hole a bit more squared and we are going to get the square data a bit more round and we will force it in and that is the way they are approaching it and they literally try to force information in, pick it up from point A to point B and there you go. And this is the pure technology side of things. That is an ideal opportunity to then step back and engage with the business when you actually find these inconsistencies and the reason there is a new source system is because the business is changing and they realise that their old system is not sufficient for where they want to go from the vision perspective. So wherever you cut it, there is a business tag in there and the people who are doing the*

> *technology solution need to comprehend what the drivers are for this new system, from the business point of view and fundamentally it comes down to this attitude of business and IT. A lot of IT just see it from an IT point of view, and they don't appreciate or recognise the fact that it is the business that is the driver and not the IT."*

This specific interpretation of *adaptability to the innate environment* led me to identify an important concept based on variations in modeling considerations in different project types and industry sectors. I call this concept a *Taxonomy of Information Systems,* or briefly, *System Taxonomy*. Similar to *Model Granularity* that distinctively described a major difference between modeling representations in agile methodologies compared to traditional methodologies, *System Taxonomy* is also a distinct concept in the analysis of data. *System Taxonomy* refers to another major difference influencing the practice of modeling. It suggests that the overall objective in the design of an information system influences conceptual models. Practitioner 4 explained how an underlying concept of *taxonomy* impacted his experience in different industry sectors and projects:

> *"Now, I should mention that in terms of my experience, I have not worked a lot in banks, so my answers might be quite different to someone who would be a lot working in banking area because banks do this [conceptual modeling] a lot more effectively than others that I have worked with… also I have worked in insurance only briefly and I have worked in banks in late 90s in configuration management more than in data. So, the areas I have worked is in transports, management systems, natural resource managements in*

department of agriculture, claims, and then other data domains… building industry information sectors, market information sectors and things like that."

Practitioner 6 also commented:

"… It is different in every business…. Every business that I worked in… I have worked prominently in telecom industry, but I have also worked in financial services…. In the telecom companies with various business organisations […] where the conceptual models were one of the legal deliverables, and an artefact before you go and do your logical modeling. In financial services industry, conceptual modeling had to be a prime deliverable as well… but depending on the project, conceptual models may not be done, I have worked in data migration projects; they don't do conceptual models! Because all they want to do is move data from one system to another system. They would not worry about conceptual model at that point of time. All they are doing is the migration of data and all they want to do is fitting to that data model or data structure if you like. So, in that case, conceptual modeling is not that useful but logical would be really good because then you have a lot of data quality issues so in one application of the system it says A, not necessarily has to be A in the other system and we need to model that. From conceptual point of view, it is the same. It is the customer here and it is the customer there but how you put in the customer name, would be different. So, that is that. It depends to the context of the project you use. But, from an application point of view, if you have a transaction system, it is very important to have a conceptual model because otherwise, if you do not document the conceptual model formally on a piece of paper, there is room for misinterpretation in different ways. For

instance, in the projects that we did not put the conceptual models on the paper, whenever we had meeting, we were talking about certain things but everyone had their own interpretation of it. Also in a reporting application, that is going to be different again and conceptual modeling is important there as well. It is a different context from a transactional system. Conceptual modeling will help. Overall, in transactional systems, conceptual models are important but not in all type of systems. Also, depending on the maturity of unit from the project management point of view, it is different how conceptual models are treated, even in companies that conceptual models are deliverables."

4.2.4.3 Distinct Identity

Some practitioners argued that a general understanding of the concept of *information*, as an abstract notion, has evolved so that *information* is now regarded as one of the most important *assets* of organisations. This evolved understanding of information, and its perception as a concrete asset in organisations, has in turn led to a new perspective on conceptual modeling. Conceptual models, in this new setting, possess a *distinct identity* as a means of representing and managing one of the most-important assets of organisations. In this light, the practices of conceptual modeling are deemed to bring value to organisations. Practitioners 5 and 7 indicated:

"Because if the business is now starting to think ok, information is an asset, how are we going to manage it rather than to think we need to capture all that transactions; it starts thinking about information as information not as transactions associated with doing business. So, what I think is happened is that the whole conceptual piece of modeling, in its theoretical positions were

valid to business for the past organisations, but it has just now arrived effectively for quite some time and actually has risen from an almost theoretical university type of stuff that is just good for contemplation type of view… it actually has real value to the organisations now […] It is actually the concepts and its level… the time has come that it has risen and has real relevance in the business context. We see with the whole rise of information management, information analytics, and information governance."

"… So, I think we are seeing a slight shift in the focus of organisations. Things like analytics, and actually knowing the market and changing the way you do business to be more versatile is having an impact on the fundamental conception of what that business does..."

4.2.5 Maturing Industry

The *in vivo* code of *Maturing Industry,* and how conceptual modeling evolves in the context of it, was described by Practitioner 5 as follows:

"… Because the business viewpoints are now rising, the technology and the way we discuss information and the level of conversation rises into that high level… and from an information-management and technology point of view… to actually be able to contribute and add value into your organisation, you have got to be able to talk in that level; in a semi-abstracted conceptual level. […] and because of that broadening of business viewpoints, maturing industry, the business attitudes to the information... Their viewpoints have risen and so have the technology and the support they would like from the technology in that space. It had to rise as well to support that and to reflect

> *that. And so, you get into that higher-level discourse and the value of conceptual and the high level logical models start to rise."*

As the concept of information as an organisational asset became more concrete and conceptual models acquired a *distinct identity* in bringing value to organisations, practitioners were proposing that IT as an industry is *maturing* in three ways. First, the industry is becoming increasingly *standardised* (which is an indication of its maturity). Second, advances in available baseline technologies are increasingly masking complexities in practices and domains. Consequently, conversations among stakeholders are changing in ways that enable them to tackle more sophisticated problems. Third, the advancement of baseline information technologies allows businesses to have a deeper impact on the design of information systems (compared to the prevailing practices in which technologists predominantly drove designs). As a result, unprecedented forms of participation in problem solving are emerging.

In the subsections below, I discuss each of these three maturing frontiers as subthemes of *Maturing Industry*.

4.2.5.1 Standardisation

Similar to many other industries, practitioners argued that IT as an industry is inevitably moving towards *standardisation*. To make reuse and interchange possible between different IT providers, they argued that *standardisation* is a necessary milestone in the effective practice of system development. Furthermore, standardisation is necessary for effective management of information as assets in organisations. Practitioners 7 and 3 commented:

"There are a myriad of standards out there; there are a myriad of best practices out there from coding, to infrastructure, to business processes, to financial regulation, to regulatory regulation, industry best practices, industry models, and they are all attempting to fine tune that enterprise around what it does and how it does it and how it recalls what it is doing. And, I think we are moving where I use an analogy back to industrial age where we had the industrial revolution and we gone from manually intensive simple solutions to machines doing these things, and everybody came up with their own design of these machines and everybody building tracks and trains but they all using different gauges and they all using different size fixtures and fittings and then as the myriad of different manufacturers started to diminish and a few key players started to evolve, they started realising that because they are doing things differently, they could not actually interchange their product with someone else. So, they then started standardise fixtures and fittings and standardise what is going to work and what is not going to work. This is 100 years ago. IT and the information age is a very recent thing, a very young industry, about 50 years max but it is bouncing on such a high pace and I think we are still going through that phase of standardisation and businesses themselves are also going through transition of producing they used to be manufacturers, or they used to be banks, or they used to be retail outlet and they were focused on physical things that they were doing and now, with the explosion of information age, organisations are finding that they can be anything they want as long as they can manage the information."

"In IT industry, we keep inventing the wheel. We have so many banks, quite large industries investing in IT and every time we go from one industry to

> *another, the problems are the same but we have to design from the scratch. [It would be ideal] that IT would be more like a car company. Ok, the car has four wheels and certain parts. How you build a part should not be your concern anymore. You should be able to grab a part, already done and plug it in your system. Creating a part should not be your system project anymore. In some areas in IT, [this has been done.] For instance, in user interface to create [a particular] kind of menu, programmers take that for granted and we are not concerned about that anymore. But there should be a higher level too. For example, in telecommunication industries, if we talk about telecom services, they should be already a design and a pattern and we should be able to use it and don't question it and reuse it in our model and unfortunately, every time we have to redesign the same."*

Nevertheless, practitioners insisted that standardisation does not inhibit idiosyncrasies in information systems. Instead, standardisation provides the building blocks for different information systems with varying qualities. Practitioner 2 explained:

> *'I think the differentiation across companies is less important. Technology standardises everyone very quickly, simply. The differences would be in qualities, the speed, etc. You have two houses and you buy bricks [to build them]… you don't need to reinvent bricks… you still get two houses by bricks, one liveable, one non-liveable but still made by bricks.'*

4.2.5.2 Facilitation of Sophisticated Dialogues by Masking Complexity

As the IT industry matures, practitioners proposed that the innate complexity of practice and system domains is increasingly embraced and *masked* by the pervasiveness of advanced information technologies. A maturing IT industry in this sense provides a technological baseline through which more sophisticated dialogues among stakeholders become possible. Practitioner 5 commented:

> *"So, in that sense, [number] one the maturity of the IS as such and two, the technical sophistication of the business too… they used to work in that type of technology and they no longer need to worry about the technical complexity. They don't want to talk to people who are worried about technical complexity. It's the maturity across the board in terms of using the technology that has changed the conversations over the years…. Yes! They can talk that language but universities and skilled people are now much more attuned to business and the sophistication and standardisation and the packagerisation of IT has made that possible, meaning that they don't have to worry about that so much because, the complexity of the technology is taken care of it. You no longer need to know the implementation pieces underneath. And you now dealing with something like a spreadsheet, which takes that abstraction and the technical complexity out and you must have to be able to speak business terms because the pure technology is no longer in such great volume for these people. So, if you come to work in IS area, you better be skilled in the business pieces or you don't have a job."*

Baseline IT technologies have made it possible for both business and technical stakeholders to divert their focus to deeper issues that previously could not have been canvassed. Practitioner 5 further elaborated on the new setting for practice:

> *"So, it is a classical case of the old saying that it is hard to remember where you are in a swamp when you are at the backside of an alligator. But when you have the support of back structure now, the base technology now, you can*

say we can now address that. We can now think about it at a higher level and it takes the threat of the alligator away to a certain extent. So, you then can see above into the whole swamp. Technology effectively removes the day-to-day transactional static from dealing with the data."

The elevation of dialogues among stakeholders has deep implications for system development practices. Practitioner 4 described the prevailing context of practice as follows:

"But the major issue that I face on a day-to-day basis is the issue of the semantic datasets map. ... So, most of the organisations I have worked in are trying to understand what type of data they actually have in terms of semantic datasets… umm… or if they do understand it! They certainly do not have a big picture of it and then the later frustration is that, there is no tool set to capture that. This is from the perspective of an information manager and a data architect. The second major frustration that I face is that modeling that is being done tends to be application driven and not data driven, with the exception of data warehouses. They tend to build their models because they can. But in most other areas, it is application driven. And in most of the organisations that I have worked in, they have absolute spaghetti of data stores, redundant and duplicate data stores, etc. So, that is another major frustration. And then the third one that has been written about for decades is the persistence of the information silos and the degree that information silos are tied to the actual politics. That is a major frustration for developers, and the larger the organisation, the more their information assets is divided in silos than are attached to organisational divisions with political alignments."

In the context of a *maturing industry,* however, more sophisticated dialogues among stakeholders are becoming possible. This is particularly due to the fact that the contribution of conceptual modeling has gone beyond simply delivering an artefact (as it was perceived in the prevailing context of practice). Conceptual modeling has gained a *distinct identity* because of its more prominent role in identifying information as organisational assets. In this regard, an elevated dialogue that takes place in the context of conceptual modeling practice seems to mitigate some of the issues indicated by Practitioner 4, including semantics of datasets and transferability of data semantic maps. Moreover, it also has a paramount role in defining what is relevant with respect to information as an organisational asset. Practitioner 4 stated:

> *"So, I think we are seeing a slight shift in the focus of organisations. Things like analytics, and actually knowing the market and changing the way you do business to be more versatile is having an impact on the fundamental conception of what that business does."*

Practitioner 5 also stated:

> *"But, we have been doing an insurance classification over and above the industry because we need this for insurance purposes. […] When we have multiple examples. Things like written premium, which is a premium in the contract that is written down and how much we made of it. Some people say that is the price that customers pay; so, that is how much they paid and it is the total amount. Some other say, it is the value customer pays after deducting GST, after we deducted government fees like terrorism levy and those sort of stuff. When we talk premium, again in the context that we include this and do not include that, we have the conversation about written premium and so, that*

> *variations, when you are trying to do information systems, the definition association and the official definition which is actually a localised variation of it is quite impactful if you will, and again it is a valid linguistics and localised vocabularies.*

4.2.5.3 Transcended Problem-Solving Methods

The impact of baseline technologies in masking complexity is not limited to how problem statements are being formulated in elevated dialogues. In addition, baseline technologies provide new platforms for alternative solutions and *transcended problem-solving methods*. With these methods, masked complexity alters modeling perspectives in a way that the modeling objectives have increasingly placed 'the users in the driving seat.' Practitioner 5 explained:

> *"And, that is part of the visionary stuff that I am doing, and it is that I don't want to model this state! I don't want to worry about how it moulds. I want two paradigms around! I want to say to the end users, here is the data! Use it however you like! You can figure out what you want to do with it and structure it far better than I can [...] Go and use it! You figure out what value might be in it, because you are an actual business driver who can find value in it. The last thing I want to do is to get the IT and information systems on your way. We provide you the technology that allows you to do that initial exploration. From whatever data you want to go and look at, you see whether there is value in it and whether there is persistency in it rather than tying up expensive IT resources and chasing windmills. IT comes in after the event. After you find that insight! [...] You put them in the driver's seat and you let them drive and from a business point of view, if they can find value fast, go and I say just*

follow them, don't put a break or block them. If they don't find the value fast… [inaudible] …. And if you are delivering and you see the value for the organisation that persist, then you can put appropriate amount of effort and funding behind it to justify it and you will never had the problem of how do I justify this, or should we do it because it pays off. The payoff will be obvious, the funding will come and the value will drive."

Practitioner 5 further elaborated on what modeling means in the new problem-solving paradigm:

"Then, when you have got approval of persistence, you put it across to IT. Because, you already know what it looks like, you can give them structure, and you have already got a context around it and what they are trying to deliver and the value that is coming […] So the whole idea of user-driven sandpit and all those sort of stuff comes forward. Effectively, we have got two dimensions if you will. There is how we deliver this stuff. The whole waterfall against agile is about how IT is delivering. I say, turn it around. Put the users in charge and just punch in through the holes. What IT should be doing is to lay down these highways. Now, whether they do that in two [inaudible]; that is purely up for conjecture. What you want to do is; is it worthwhile to put a rail through? Because if it is; then you can work out the rest. […] It is the case of what do I deliver. Do I deliver big monolithic pieces like freeways or do I deliver series of bypaths? […] We want people to move from A where information is. What we don't want to do is to constrain people's thinking. So, you have got to allow that and in fact, encourage people to try other pathways."

4.3 Summary

In this chapter, I presented the findings from my study. First, I explained how by adopting a thematic analysis method I developed a thematic map of data. I then discussed how I identified five overarching themes. Based on principles of interpretive field studies, I elaborated on each of these themes by juxtaposing multiple views of practitioners and reflecting on the meaning of the data I collected. Table 4-3 summarises the themes and subthemes that I identified.

Table 4-3 Research findings based on themes and subthemes

Themes	***Subthemes***	***Description***
Cognitive Engagement	*Common understanding* *Coding as sense making* *Communication* *Ownership* *Translation*	Discusses practitioners' main objective in agile modeling practices is to maximise cognitive engagement of all stakeholders in order to obtain better understanding of the domain. To achieve this outcome, modeling methods in agile practice focus on attracting stakeholders' participation and deep engagement in domain semantics elicitation. They seek to achieve this goal by creating a sense of ownership and common understanding among different stakeholders, using the process of coding as a sense-making process and aiming at improving communication and translation among different world-views.
Fragmentation	*Technological fragmentation* *Theoretical fragmentation* *Solution semantics fragmentation*	Discusses three types of disconnect in the context of practice that need to be overcome in the process of modeling. One disconnection arises from discontinuities in technological infrastructures. A second disconnection arises between theories of conceptual modeling and the settings of real-life practices. A third disconnection arises from the difference between eliciting a set of system requirements and providing a solution semantics model based on business needs.
Volatility	*Lack of continuity in stakeholders* *Uncertainty due to human agency* *Politics* *Lack of paradigm fit – Need for discordant view*	Describes the highly uncertain context of practice based on four elements in which the first three elements are directly related to humans' impact in organisations. The fourth element indicates the importance of having a discordant view. If a modeling approach does not incorporate all perspectives, system volatility may be an outcome.

Living organisms	*Complexity* *Adaptability* *Distinctive identity*	Describes how conceptual models evolve through the process of system development as if they are living organisms. For conceptual models to be living organisms, they extract information about domain semantics and thus possess distinctive identities. They also constantly adapt to a complex setting as living organisms do.
Maturing Industry	*Standardisation* *Facilitation of sophisticated dialogues by masking complexity* *Transcended problem-solving methods*	Indicates how maturation of information technologies provides a setting in which the significance of conceptual modeling practice is increasing as 'information' receives greater recognition as a business asset. Standardisation of many artifacts frees developers from basic implementation burdens. It facilitates more sophisticated discourse about a domain and use of some transcended problem-solving methods.

5 CONCLUSIONS

This chapter concludes the thesis. First, I provide a recapitulation of the research and address the initial research questions based on the research findings. Second, I discuss the contributions of my study and its implications for research and practice. Finally, I analyse the strengths and limitations of my research before I discuss some prospects for future research.

5.1 Reprise

The initial motivation for conducting this exploratory study was to understand how the theoretical formalisation of conceptual modeling based on a theory of ontological expressiveness and logical completeness is related to the real-life practices of system development. In particular, I set out to explore the practice of conceptual modeling in agile methodologies as an alternative approach to conventional practices of system development using waterfall methodologies. I sought to address two research questions:

> **Research Question 1** - To what extent is conceptual modeling done when agile methodologies are used to develop information systems?

> **Research Question 2** - What are practitioners' views about the need for and importance of conceptual modeling when they use agile methodologies to develop information systems?

To answer the proposed research questions, I conducted an exploratory interpretive field study. I gathered data by interviewing eight practitioners who had significant experience in system development projects. I used thematic analysis and a hermeneutic approach to interpret the data collected from semi-structured interviews.

As a result, I obtained insights on three major aspects of context, method, and the interplay between context and method in real-life modeling practices.

> First, I identified contextual elements of modeling in real life. It became clear that the real-life practices of modeling are conducted in a context that is marked by *Fragmentation and Volatility*.
>
> Second, I discerned methods for eliciting and evaluating system requirements in agile practice. *Cognitive Engagement* of all stakeholders was distinguished as a focal activity in agile methodologies. The findings indicated that in agile methodologies, the emphasis is less on pre-determined methods to develop system artifacts and more on methods that maximise stakeholders' engagement through improving ownership, communication, and sense making.
>
> Third, it became clear that the interplay of a *fragmented* and *volatile* context with methods that are concerned with the *cognitive engagement* of all stakeholders has resulted in the conception of conceptual models as *Living Organisms* in a *Maturing Industry*. Furthermore, I saw how the interplay of context and method in agile modeling practices addresses the problem of

domain simplification in highly complex and uncertain domains. In these types of domains, modeling *adaptability*, *facilitation of sophisticated dialogues by masking complexity,* and use of *transcended problem-solving methods* address the concern over potential modeling-related simplifications of domain semantics. An evolutionary perspective of conceptual modeling is reflected under *Living Organisms* and the context of a *Maturing Industry*. It paints a background of conceptual modeling practice that embraces complexity, uncertainty, and volatility by building up better understanding of domains, *gradually,* to incorporate increasingly semantic sophistications.

By way of producing these insights, this exploratory study has contributed to the under-researched areas of conceptual modeling context, method, and their interplay. Furthermore, it has provided some explanations on why the evolutionary perspective of conceptual modeling leads to alternative methods for conceptual modeling.

5.2 Findings

In addressing the two proposed research questions above, my study suggests that practitioners agree on the growing importance of conceptual modeling in alternative and new system development methodologies (such as agile). The practitioners I interviewed argued this outcome is occurring for two reasons. First, information is now perceived as a concrete asset that brings value to organisations. Practices that enable this asset to be extracted therefore become more important. Second, technical advances have taken away many cognitive burdens related to basic system implementation issues. As a result, practitioners can focus on understanding domain semantics rather than having to deal with technical problems.

Nonetheless, while practitioners argued that conceptual modeling was becoming more important, they indicated that conceptual modeling in traditional system development methodologies differs substantially from conceptual modeling in the practices of agile methodologies. These differences manifest in two ways: *Model Granularity* and some awareness to *System Taxonomy*.

In agile methodologies, practitioners indicated that conceptual models are often high-level models that represent domain-related subject matters in coarse grains. This practice is contrary to the practice of conceptual modeling in waterfall methodologies. In waterfall methodologies, conceptual models represent domain semantics in detail to provide a basis for design. In other words, the representation in waterfall methodologies is fine grained. In agile methodologies, however, conceptual models evolve throughout the system development process. Modeling of domain semantics is not a design prerequisite for system development. Therefore, while practitioners agree that conceptual modeling is increasingly important in system development, they reject the need to develop *a priori*, fine-granularity models to represent domain semantics.

Moreover, practitioners stressed that the overall objectives of information systems influence conceptual modeling practices. While conventional conceptual modeling practices are often unresponsive to differences among the objectives of information systems, practitioners reported that agile methodologies take these differences into account. In particular, the granularity of conceptual models prepared in agile methodologies depends on the overall objectives of a system. Practitioners elaborated on some of these differences in relation to different industry sectors and their specific requirements. For instance, the banking and finance industry usually has high demands in terms of accuracy and reliability. However, the education industry often has the objectives of boosting creativity and lateral thinking among system users. While specific guidelines about the needs of different industries do not exist, practitioners reported that they have experienced differences in the approach that different industries adopt to conceptual modeling procedures.

To sum up, the results of this study indicate that conceptual modeling is not a clear-cut process that commences and terminates prior to all other activities in the system development process. Instead, it is an evolving process that may continue throughout the development of information systems. This outcome conforms to some propositions in the existing literature. For instance, Roussopoulos and Karagiannis (2009) assert that a semantic domain schema should evolve continually as other system development activities proceed. This perspective on the nature of conceptual modeling practice was particularly evident in the *Living Organisms* and *Maturing Industry* themes, where practitioners underlined the evolutionary nature of conceptual modeling practice.

In light of these outcomes, it is important to understand whether and how these results inform current conceptual modeling theory and practice. In other words, the implications of these outcomes must be identified. In the following subsections, I discuss some implications for theory and practice.

5.2.1 Implications for Theory

Extending on Wand and Weber's ontological framework of conceptual modeling (1988, 1990, 1993, 1995), Burton-Jones (2012) elaborates on Representation Theory as a theory of "'core' information systems phenomena" (p. 2). Based on his reading of the ontological framework and explicit statements in the literature, he identifies five major assumptions that underpin Representation Theory. I provide a summary of these assumptions in Table 5-1.

Table 5-1 Representation Theory Assumptions adapted from (Burton-Jones, 2012, p. 3)

Underpinning Assumptions of Representation Theory
1. Information systems are representational artifacts.
2. The representations provided by an information system are provided via three system structures: surface, deep, and physical.
3. People desire faithful representations of domains of interest to them because more faithful representations provide a firmer basis for action than unfaithful representations.
4. Any representation will be a partial and fallible reflection of the real-world domain it is supposed to reflect.
5. Tokens (data) populate the system's deep structure, and by so doing inherit the meaning specified in that deep structure.

By identifying the underlying assumptions of Representation Theory, Burton-Jones (2012) illustrates how applying an ontological perspective to conceptual modeling has led researchers to ask specific questions about conceptual modeling.

Having adopted the ontological framework of conceptual modeling as the theoretical lens in this qualitative study, I follow the same tradition that is elaborated by Burton-Jones (2012). In particular, I argue that the findings of my study are related to assumptions 3 and 4 in Table 5-1.

As consistently emphasised by practitioners, a pressing need to bring value in a timely fashion to organisations impacts the practice of system development, including conceptual modeling practice. The existing literature also emphasises the significance of timely delivery of information systems to bring competitive advantage to organisations (Turk et al., 2005). Nonetheless, practitioners in this study constantly underlined the role of *Volatility* and *Fragmentation* in the context of real-life modeling practices. Although a faithful representation of domain semantics is desirable (assumption 3), in the presence of uncertainty in highly complex domains, they explained that representational fidelity is not attainable within the timeframes available for system development projects. Therefore, in line with assumption 4 in Table 5-1, partial and fallible representations of domain semantics provide the design basis in system development projects.

In the context of agile methodologies, however, assumptions 3 and 4 of Representation Theory motivate a finer question about conceptual modeling methods. Specifically, an evolutionary perspective of conceptual modeling undermines *a priori* specification of system requirements as a basis for design. In this light, a new question can be formulated about conceptual modeling in information system development:

> *In what ways do practitioners overcome the difficulties that arise with partial and fallible domain representations so that the resulting system is a better fit to the design objectives?*

This question is interesting because it implies a link between two main findings of this research: *Model Granularity* and *System Taxonomy*. It indicates that the impact of the overall methods that are used in system development projects must be understood with respect to the specific objectives of design in different information systems. In the context of real-life practices in particular, measures of *Model Granularity* in conceptual models might be a function of system objectives and therefore *System Taxonomy*. Moreover, because impartial and infallible domain representations are impossible to achieve, practitioners tailor the fidelity of a domain representation based on the specific objectives the information system is designed to fulfill.

This reading of my findings (in the light of Representation Theory) shows that current guidelines for conceptual modeling need to be formally and theoretically linked with *System Taxonomies*.

A second implication for theory based on the findings arises from the concept of *Model Granularity* and assumption 4 of Representation Theory. On numerous occasions, practitioners emphasised that the purpose of conceptual modeling in agile methodologies is not to provide a complete basis for design because *a priori* specification of complete system requirements is impossible. Instead, practitioners seemed more concerned with devising methods that enhance the *cognitive engagement* of stakeholders. In that vein, instead of focusing on modeling scripts that supposedly were complete, practitioners focused on maximising stakeholders' engagement with the practice of domain understanding to yield a better design.

From this perspective, less reliance on documentation in an agile iterative approach is understandable. Indeed, based on the notion of *Model Granularity*, a theoretical issue would be to see whether high-level modeling practices that are undertaken in agile methodologies have led practitioners to focus on modeling information systems artifacts more, or whether they helped them to concentrate on the elicitation of the underlying domain phenomena.

Last, based on the findings of this study, formalising a theoretical understanding of the evolutionary perspective to system development projects based on TOE and TLC seems a fruitful area for further research. In particular, it would be useful to understand the ontological underpinnings of the iterative approach in terms of modelers' use of conceptual modeling grammars and scripts. For instance, as high-level modeling seems to be the introductory step in agile methodologies, researchers might examine whether choice of subject matter in high-level modeling can be explained by ontological constructs.

5.2.2 Implications for Practice

The findings of my study have two implications for practice. First, they show that the practice of conceptual modeling is not becoming obsolete. Rather, my findings indicate that conceptual modeling is becoming increasingly important in the context of agile system development methodologies. Nonetheless, little theory exists to inform the practice of agile modeling in real-life projects. To address this concern, my study extends the application of Representation Theory to the new setting of agile methodologies. It provides qualitative support for using an ontological perspective of conceptual modeling to design modeling guidelines for agile methodologies.

Second, obtaining a deeper understanding of the notion of *System Taxonomy* should enable modeling guidelines that better fit the context and purpose of information systems to be derived. As described earlier, practitioners reported different approaches to the practice of modeling in different industries. While all information systems primarily seek to represent domain semantics, the purpose of an information system affects how conceptual modeling is undertaken. Because the potential differences in objectives across different types of information systems have not been recognised, incomplete modeling guidelines exist. Nonetheless, the question of whether differences arise between representations of information systems based on their overall objectives requires further research. Such research would assist in determining whether current conceptual modeling guidelines that are based on TOE and TLC need to be refined.

5.3 Contributions

As an interpretive study, the contributions of this research must be evaluated based on a notion of *generalisability* (Walsham, 1993). In qualitative studies, contrary to quantitative studies, the representativeness of the results is not supported statistically. Instead, in interpretive studies contributing to the body of knowledge is based "on the plausibility and cogency of the logical reasoning used in describing the results from the cases, and in drawing conclusions from them" (Walsham, 1993, p. 15). Walsham (1993) describes generalisability as identifying *generative mechanisms* or *tendencies* in data. He identifies four types of generalisations as potential contributions in interpretive studies: "development of concepts, generation of theory, drawing of specific implications, and contribution of rich insight" (Walsham, 1993, p. 79).

While these four types of generalisation have overlaps in the way they make contributions to the body of knowledge, I argue that my study makes contributions particularly in two of the named areas—through the development of new concepts and the contribution of rich insight.

> <u>*Development of new concepts:*</u> I found that *Model Granularity* emerged as a major concept in describing the differences between conceptual modeling practices in agile and waterfall methodologies. Other concepts that emerged as a result of the interpretive and thematic analyses were also important—for instance, *System Taxonomy*, *Coding as Sense Making*, and *Lack of Paradigm Fit*. However, these other notions were either extending some concepts that prior literature had already identified or were enriching them. For instance, the concept of *System Taxonomy* extends the discussions initially developed by Davis (1982) to recognise *uncertainty* as a contextual element that influences

elicitation of system requirements. Similarly, *Coding as Sense Making* was inspired by Snowden's work (Kurtz & Snowden, 2003; Snowden, 2002) in relation to complexity in knowledge management. Last, the notion of *Lack of Paradigm Fit* was influenced by Parsons' work (Lukyanenko & Parsons, 2011a, 2011b, 2013a, 2013b; Parsons & Wand, 2000, 2008) concerning the impacts of abstraction and classification on modeling.

To the best of my knowledge, however, the notion of *Model Granularity* has not been examined in the existing conceptual modeling literature. As discussed earlier under Section 5.2, *Model Granularity* frames the major difference between modeling practices in agile and waterfall methodologies. As a consequence, recognising this major difference in the form of a new concept based on the findings of my study arguably enriches the literature. It also motivates work to tease out new and deeper questions about the core phenomena of conceptual modeling practice (as discussed in detail in Section 5.2).

Contribution of rich insight: My study contributes to an in-depth understanding of conceptual modeling in practice. The following paragraphs summarise the major insights based on my findings.

First, conceptual modeling is not always a visible and distinct practice in the early stages of system development. Instead, it may develop concurrently as system development proceeds. This finding reinforces adoption of an evolutionary perspective on the practice of conceptual modeling, particularly in the context of volatile, highly complex domains. Understanding an evolutionary perspective based on the current ontological framework of conceptual modeling or a theoretical extension of this framework is a matter for further research.

Second, conceptual modeling is increasingly recognised as bringing value to organisations as a *distinct identity*. In the process of conceptual modeling, information about business domain semantics is extracted by integrating domains' expert knowledge. This information is now considered to be an organisational asset.

Third, the findings of my study strongly support the statement that highly volatile contexts have a significant impact on the methods practitioners use to develop information systems. In this regard, the concept of *System Taxonomy* emerged as having an important impact on the practices used during conceptual modeling. This concept suggests that two factors should be considered if conceptual modeling guidelines are going to accommodate different types of information systems based on their overall objectives: (a) the context of practice in terms of *fragmentation* and *volatility*; and (b) the overall objectives of an information system design. Formalising the impact of these elements based on a theoretical framework is also a subject for further research.

5.4 Strengths and Limitations

Scholars have long debated measures of validity for qualitative studies (Creswell & Miller, 2000; Denzin & Lincoln, 2000; Lincoln, Lynham, & Guba, 2011; Miles & Huberman, 1994; Myers & Avison, 1997). Nonetheless, a consensus exists about the fundamental differences between criteria for qualitative and quantitate research.

Lincoln, A., and Guba (2011) argue that internal validity, external validity, reliability, and objectivity are the criteria used to judge the quality of quantitative research, while credibility, transferability, dependability, and conformability are the criteria used to judge the quality of qualitative research.

In Guba and Lincoln's (2011) framework, credibility is assessed in terms of the believability of the results, while transferability is assessed in terms of the extent to which the results of the study can be generalised or transferred to other settings. Similarly, dependability and confirmability of the research are assessed in terms of the extent to which the context of the results is clearly described (dependability) so that by contextualising the results they can be corroborated by other researchers (confirmability).

In giving an evaluation of the validity and credibility of the results of this study, I reflect on the seven principles of interpretive research provided by Klein and Myers (1999). In particular, I discuss the strengths and limitations of my research from three perspectives—namely, research methodology, theory, and data collection. In doing so, I pursue two fundamental goals.

First, I evaluate the results of my study based on The Fundamental Principle of the Hermeneutic Circle (as the foundation principle for interpretive hermeneutic research). In this way, I demonstrate how my use of a hermeneutic circle informed the results I obtained from the thematic analysis of the data obtained in my research.

Second, by addressing seven principles of hermeneutics in relation to the results of my study, I adopt a framework to link my arguments about the strengths and limitations of this research to the two other aspects that I evaluate for theory and data collection.

Table 5-2 provides a summary of the strengths and limitations of my research.

Table 5-2 Strengths and Limitations of the study

		Discussion
Theory	**Strengths**	A well-grounded theoretical lens based on empirical evidence from one of the longest-running programs of research in Information Systems. Extending application of the Ontological Framework of Conceptual Modeling to an interpretive field study.
	Limitations	Bounded by Bunge's ontological framework in considering 'entities' as the primary ontological units—no theoretical triangulation
Research Methodology	**Strengths**	Complying with the hermeneutic circle to make sense of conflicting assumptions and data in a cohesive whole through credible inferences. Reinforcing "people as producers not just products of history" by conducting semi-structured interviews to obtain practitioners' views about status and significance of conceptual modeling. Producing deep insight into the phenomena of information system development. Adopting thematic analysis as data analysis method provided a framework to particularly demonstrate how themes of data emerged as a result of dialogical reasoning in complying with Klein and Myers' (1999) principle five
	Limitations	Not contextualising the findings based on participants' social and political backgrounds. Absence of a Social Critical Perspective in data analysis. Narrower range of perspectives obtained because I interviewed practitioners who were advocates of agile methodologies.
Data Collection	**Strengths**	Obtaining in-depth insight by using semi-structured and face-to-face interviews as my data collection method. Minimised elite bias by interviewing practitioners from diverse backgrounds.
	Limitations	Not having interviewed any critical case Not having interviewed any female practitioners. No data collection method triangulation.

5.4.1 Evaluating Research Methodology - Principles of Interpretive Field Study

As Klein and Myers (1999) point out, while not all seven principles of interpretive field studies may be applicable in all research settings, nonetheless their application is not arbitrary. In the following paragraphs, I elaborate on how I attempted to comply with each of their seven principles. I also explain the limitations of my study based on these principles before linking them to limitations of theory and data collection method in the next subsection.

1. *The Fundamental Principle of the Hermeneutic Circle:* Klein and Myers (1999) argue that "the idea of the hermeneutic circle suggests that we come to understand a complex whole from preconceptions about the meaning of its parts and their interrelationships [and that] the terms "parts" and "whole" should be given a broad and liberal interpretation" (p.71). I provided an initial broad interpretation for this concept in Figure 3-1, by depicting how I understood and interpreted each data extract as a part, in the context of its respective interview, and the entire dataset as a whole. Similarly, in complying with this principle, I argued that each interview in its own right constituted a part, but it also had to be understood in relation to the entire set of interviews as a whole.

   Furthermore, as a result of iterations between parts and whole, a new understanding of the role of conceptual modeling in agile practices emerged. This new understanding also principally evolved through iterations between partial understanding of some literature and a broader theoretical understanding of the role of conceptual modeling as a whole.

A partial understanding of conceptual modeling, which was also supported by some of the data I obtained, implied that the practice of conceptual modeling is irrelevant in the context of agile methodologies. This partial understanding was based on the literature. It arose based on the *presumption* that conceptual modeling practice is a visible, distinct practice that takes place during the initial stages of system development. This presumption also implied that specific conceptual modeling scripts using specific conceptual modeling grammars ought to be generated as a result of the practice of conceptual modeling.

However, this presumption was clearly ruled out in the context of agile practice. Practitioners reported that most often extensive and detailed conceptual modeling scripts are not prepared. Furthermore, they questioned the perceived benefits and feasibility of practices that rely on detailed conceptual modeling scripts as providing a basis for design. Instead, their narratives reflected a major theme, which I called *Cognitive Engagement,* whereby the involvement of stakeholders was the focal activity in the practice of system development. As a result of practicing *Cognitive Engagement*, specific conceptual modeling scripts may or may not be generated. However, the emphasis is not on the conceptual modeling scripts because better domain understanding is achieved through *Cognitive Engagement* of stakeholders and not through the generation of specific scripts.

Because the presumption of visible and distinct conceptual modeling scripts was rejected through the emergence of the *Cognitive Engagement* theme, I identified a new focus for another hermeneutic circle. In the new hermeneutic circle, conceptual modeling practice was understood as an evolutionary process that occurs throughout the process of system development. This understanding of conceptual modeling was guided by a broader theoretical understanding based on the Theory of Representation for the role of conceptual models in improving domain understanding. In light of the evolutionary perspective of conceptual modeling, its role was neither obsolete nor controversial. The new hermeneutic circle particularly made sense in light of some other findings of my study—specifically, that conceptual understanding of a domain's semantics was becoming increasingly important in the context of practice.

2. *The Principle of Contextualisation:* Klein and Myers (1999) argue that a fundamental task in an interpretive study is to make sense of the phenomena studied in their context. However, different contexts apply to each phenomenon. In this study, similarly, different contexts had to be explored in relation to the focal phenomena. For instance, I could have explored practitioners' background as informants in this research to see how their background influenced their perspectives and conduct in practice. While I believe not having included such an analysis is a limitation of my research, I have attempted to comply with *The Principle of Contextualisation* by looking at how system development methodologies have evolved across time. In this regard, I contextualised exploring real-life practices of conceptual modeling in the setting of system development methodologies.

To attain this contextualisation, I chose a narrative that problematised conventional assumptions about conceptual modeling. In particular, I adopted a historical view of the development of system development methodologies. For instance, in Chapter 2 I argued that the preliminary framework of the ontological perspective of conceptual modeling is influenced by sequential methodologies. I indicated that alternative system development methodologies and new types of information systems are emerging (painting a historical development of methodologies). The conventional assumptions that might be taken as legitimate statements about these methodologies therefore need to be reassessed.

Furthermore, by choosing to interview practitioners about 'their view' of conceptual modeling, I demonstrated that I perceive people as "producers and not just the products of history" (Klein & Myers, 1999, p. 74). This is another indicator of complying with Klein and Myers' (1999) second principle (contextualisation). In short, by complying with principle 2, I attempted to incorporate the real-life perspectives of expert practitioners, as producers of the future, in our current theoretical understanding of conceptual modeling.

3. *The Principle of Interaction Between the Researcher(s) and the Subjects:* As a result of my interactions with practitioners, my understanding about conceptual modeling evolved considerably during the course of my research. The hallmark of this change occurred during the pilot interview.

   My preliminary idea in terms of this research was to find out whether conceptual modeling practice had become irrelevant in agile methodologies.

Furthermore, I was interested to find out what perspectives informed domain understanding in agile methodologies, if agile practitioners claimed conceptual modeling was no longer applicable.

After conducting the pilot interview, I realised using a singular approach to try to understand conceptual modeling in practice would not be fruitful. Indeed, I recognised that I had to be flexible about different interpretations that different practitioners might provide of what constitutes conceptual modeling practice.

As a result, I stepped away from depicting the conceptual modeling status in practice based on two hypothetical extremes—namely, conceptual modeling is either no longer useful or it continues to used in the same way it is used in traditional methodologies. Furthermore, I discarded ideas about undertaking a detailed investigation of potential theoretical counterparts for TOE and TLC in agile practices. Rather, I began to redesign the research questions so that my study could incorporate many forms of understanding about the practice of conceptual modeling.

In the revised form of my research enquiry, I allowed for different interpretations of conceptual modeling, providing the focus of an interpretation was to improve domain understanding. This new perspective was motivated by my interactions with practitioners (principle 3), the hermeneutic circle (principle 1), and the pivotal role of theory (principle 4). This circle of interpretation became possible when I adopted a broader definition of conceptual modeling based on Representation Theory—one that focused on improving domain understanding as a whole rather than providing specific scripts for design.

4. *The Principle of Abstraction and Generalisation:* I addressed this principle, primarily under Section 5.3, when I discussed the contributions of my research to be *developing new concepts* and *obtaining rich insights*. Furthermore, the ontological perspective of conceptual modeling and the literature overview in Chapter 2 provided the generalised conceptions that allow inferences from the idiosyncratic narrations of individual practitioners.

   For instance, as I explained above under the first and third principles of interpretive studies, through the lens of the theoretical framework of conceptual modeling I was able to envisage better domain understanding to be the main focus of conceptual modeling practice in agile methodologies (instead of providing conceptual modeling scripts as a basis for design). Without a theoretical understanding of conceptual modeling, I doubt I could have made sense of the many data extracts, indicating the importance of communication, ownership, coding as sense making, and translation under the theme of cognitive engagement. Without such a theoretical lens, similarly, I doubt I would have moved beyond presumptions of visible conceptual modeling practice. In this regard, the ontological perspective of conceptual modeling and the Theory of Representation provided the concepts and background that I needed to undertake cogent and plausible reasoning about the data.

5. *The Principle of Dialogical Reasoning:* The details reported in Chapter 4, where I discussed development of the thematic map of data under Subsections 4.1.1 to 4.1.3, are one manifestation of the dialogical reasoning between myself, as the interpreter, and the data as the text. Furthermore, by contextualising an understanding of conceptual modeling in the settings of system development methodologies, including waterfall (through a review of literature), and agile methodologies (through a review of literature and, in particularly, the conduct of semi-structured interviews), I attempted to reveal the underlying assumptions, or the prejudices, that shape each of these methodologies.

   As a result of this process, my perspective about conceptual modeling also shifted. This outcome occurred as I applied the hermeneutic circle to my data. It also occurred as I reflected on the historicity of the practice of conceptual modeling (as discussed in Chapter 2). In particular, I began to understand traditional system development methodologies based on notions of line manufacturing that were prevailing in the seventies (Royce, 1970; Sumrell, 2007; Thummadi, Lyytinen, & Berente, 2012), which eventually led to the evolutionary and socio-technical perspectives of agile methodologies in the current times (Abrahamsson et al., May 2003; Ambler, 2014; Beck et al., 2001; Erickson et al., 2005; Highsmith, 2002; Highsmith & Cockburn, 2001; Nerur & Balijepally, 2007; Turk et al., 2005).

   During the course of my analysis based on the historicity of methodologies, I revisited my presumptions about conceptual modeling methods that would fit all system types and objectives. As understanding of conceptual modeling practice matured over time, I noted that the presumption of one size fits all no longer held. This transition occurred alongside the introduction of more diversified types of information systems enabled by advances in technology, Therefore, I dropped my prejudice about visible and distinct conceptual modeling scripts providing a basis for design. I also dropped my prejudice about conceptual models remaining unaffected by the types and objectives of the designed system.

Another instance of complying with the dialogical reasoning principle is related to the theme of *Maturing Industry*, discussed under Subsections 4.2.5.2 and 4.2.5.3. As every practitioner consistently emphasised the complexity of domain semantics, no plausible reasoning was initially available to explain how the simple, high-level modeling often used in agile methodologies could be effective. Using dialogical reasoning based on my data, however, I came to understand that improved technologies now enable practitioners to focus on the concepts related to domain semantics rather than system design and implementation issues.

6. *The Principle of Multiple Interpretations:* This principle is reflected throughout this study in the narrative presenting the thematic view of the data I collected as well as my hermeneutic interpretation of the data. I provided a complete account of multiple interpretations of conceptual modeling definition and role from practitioners' perspectives in Chapter 4. These views were wide-ranging—for instance, Practitioner 8 regarded conceptual modeling (particularly in its traditional form) as unimportant, whereas Practitioners 3 and 6 regarded high-level conceptual modeling as

critical. Nevertheless, using the principles of hermeneutic interpretation, these multiple viewpoints could be placed in a cohesive frame.

Nonetheless, my study achieves only limited compliance with principle 6 because I interviewed only a small number of practitioners. On the one hand, I reached saturation with my interviews. On the other hand, I did not examine critical cases in my study. While multiple interpretations are incorporated in the collection and analysis of data, the practitioners predominantly showed general acceptance of alternative methodologies. Moreover, none offered diehard critiques of agile methodologies.

Although I targeted a diverse pool of participants with my sampling method and in spite of reaching saturation, I envisage that longer engagement in the field might have provided an opportunity to interview practitioners who had more diverse perspectives on system development methodologies. In particular, my results might have been enriched if I were able to interview practitioners who held strong contrary views on the merits of agile methodologies.

7. *The Principle of Suspicion*: I particularly applied the principle of suspicion during the interviews I conducted. By adopting a critical perspective on the responses that practitioners were providing, I was motivated to use probing questions to go beyond the surface of the initial positions they espoused. Also, in the course of data analysis, I juxtaposed alternative practitioner viewpoints and different theoretical perspectives. As a result, I found that *common information models,* also known as *canonical models* or *reference models*, entail similar implications for practice as conceptual models.

Therefore, an argument that conceptual models are not relevant in agile methodologies but common information models are relevant reflects confusion in terminologies. By adopting a critical position and using probing questions during my interviews, I uncovered the nature of this confusion.

Complying with principle 7, *The Principle of Suspicion*, also entailed complying with *The Principle of Multiple Interpretations* and *The Principle of Interaction Between The Researcher and The Subjects.* The reason is that the practitioners and I ultimately started to see each other's perspectives about the practice of conceptual modeling.

I did not apply principle 7, however, from the perspective of a Social Critical Theory (Klein & Myers, 1999). Adopting a social critical perspective primarily entails using methods of critical analysis rather than interpretivism. As another limitation of my research, I recognise that the social and political backgrounds and personal interests of the practitioners I interviewed could have substantially influenced the responses they provided.

As far as the interdependence of the principles entails, the pivotal role of principle four and the theory in extending all other principles was paramount. This observation accords with Klein and Myers' (1999) prediction about the role of theory in interpretive studies. Without hermeneutic iterations between parts and whole (principle 1) and a solid theoretical framework (principle 4), I was not able to gain any awareness about my own historicity as a researcher (principle 3), neither I could make cogent inferences to reflect on principles five, six, and seven. In addition, adopting thematic analysis as a complementary data analysis method made possible explicit elaborations on principles three (interaction between researcher and subjects) and five (the principle of dialogical reasoning). Given that Klein and Myers (1999) underscore the importance of demonstrating principles three and five in interpretive research, I argue that my use of thematic analysis is a strength of my research because thematic analysis provides a framework to show how my analysis developed and how each theme emerged.

5.4.2 Evaluating Choice of Theory

Choosing the Ontological Framework of Conceptual Modeling as the theoretical lens strengthened my research by providing a well-founded theory and a sharp perspective on the phenomena I was studying. As discussed in Chapter 2, this framework is one of the longest-running programs of research in Information Systems, and it is based on rich theoretical concepts and a large amount of empirical evidence. Nonetheless, like any theoretical framework, the ontological perspective of conceptual modeling is constrained by its own specific assumptions and perspective on reality.

In this regard, Weber (1997) discusses these assumptions in four areas: ontological assumptions, epistemological assumptions, social-context assumptions, and representation assumptions. Under the ontological perspective of conceptual modeling, he explains how information systems are *representations* of real-world domains. He also provides a detailed view on how the ontological framework of conceptual modeling initially embraces *functionalism* but goes beyond the assumptions of this paradigm.

In the paradigm of functionalism, the primary goal is to build theories that account for *order* in an *independent world,* using *positivistic research methods*. Holding the position of a critical and scientific realist, Weber (1997) first focuses on how the functionalist philosophical paradigm underpins the ontological framework of conceptual modeling. Simultaneously, he also discusses how an ontological framework of conceptual modeling based on Bunge's ontology can accommodate other philosophical paradigms such as interpretivism and neohumanism[*] when their concerns fall within the domains of ontological theory. From this perspective, conducting my interpretive research based on an ontological framework of conceptual modeling contributes to knowledge about the framework by showing how it can be used in a non-functionalist (interpretivist) paradigm.

Nevertheless, as Bunge's ontology underpins this theoretical framework, this study is constrained by Bunge's ontology. For instance, Bunge's ontology perceives 'entities' to be the primary ontological units of the world, while other ontological frameworks do not consider 'entities' to be the primary ontological units. In a *relational ontology*, for instance, 'phenomena' are considered to be the primary ontological unit (Barad, 2014; Orlikowski & Scott, 2008; Scott & Orlikowski, 2014). Further research could examine the implications of holding alternative views about the primary ontological unit on conceptual modeling.

---

[*] Based on Hirschheim, Klein, and Lyytinen (1995), *interpretivism* is the philosophical paradigm of social relativity. In this paradigm, interpretive research methods are the primary methods used to account for order in a world that is constructed by the subjectivity of humans' perceptions. *Neohumanism* instead focuses on empowering individuals to reach their full potential by mitigating the effects of social and communication barriers and injustice.

5.4.3 Evaluating Method of Data Collection

By adopting semi-structured interviews as my data collection method and using probing questions, I had the opportunity to seek in-depth insights about the practice of conceptual modeling. Moreover, by using an open-ended script (semi-structured interview) in a face-to-face interview setting, I could improvise in light of practitioners' different attitudes toward the subject matter, which enriched my data collection.

As I indicated earlier, a limitation of my research is that I do not have critical cases in my interviews. While my participants had varied experience of conceptual modeling, none held strong negative views about agile methodologies. Furthermore, no female practitioner volunteered to be interviewed in my study.

Therefore, while I argue that my research reflects reasonably diverse voices of practitioners based on their collective experience in system development and design, role, nationality, industry sector, and hierarchy in organisations, having the perspective of critical cases and female participants would have enriched my data collection. Nevertheless, because I sought to obtain diverse organisational backgrounds for participants in my research, I argue that an elite bias (Myers & Newman, 2007) should have been minimised in this study.

In summary, and based on the extensive reports provided in Chapters 3 and 4, I argue that I complied with the seven guidelines provided by Myers and Newman (2007) for qualitative interviews:

First, *I situated myself as an actor* and *minimised social dissonance* by briefing practitioners about my research and myself through emails (enclosing the Explanatory

Statement in Appendix A), by conducting over three hours of a pilot interview with a lead practitioner in the field as a main point of contact for snowball references, and by attending a conference in person.

Second, I interviewed a diverse group of practitioners recruited through different sampling methods (as described in Table 3.1). In this way, I attempted to *represent a variety of voices* within the timeframe of this research. The span of time for this interpretive study had been the equivalent of 12 months full-time work, and I have had over 11 hours of direct contact with practitioners in the conduct of my interviews.

Third, by framing this research as an interpretive study (as explained in Section 5.4.1) and using thematic analysis of data (as explained in Chapter 4), I attempted to make sense of a complex practice in its own setting, recognising that each participant in this research, including myself, *is an interpreter*. Furthermore, by using probing questions and *mirroring the answers,* I refrained from imposing my views during interviews and attempted to expand the interviews based on the practitioners' language. Using *in vivo* codes in reporting the results is in particular, a reflection on complying with this principle. Because *in vivo* codes by definition incorporate informants' perspectives into the research directly by using the exact terms participants expressed. Searching for themes in the data, obtained from interviews, further reinforced the idea of "focus[ing] on common, vividly-held events and stories" (Myers & Newman, 2007, p. 17).

In summary, using the criteria explicated by Creswell and Miller (2000), my study establishes its credibility in six ways:

1. Through *triangulation* of data analysis methods by adopting thematic analysis and Klein and Myers' (1999) principles of interpretive field studies and triangulation of data obtained from multiple participants.
2. By searching for *disconfirming evidence* through adopting the fundamental principle of the hermeneutic circle and principle of dialogical reasoning.
3. By attending to *researcher reflexivity* through adopting the principle of suspicion.
4. By seeking *collaboration* from practitioners through extensive interviews and incorporating participants' perspectives in the design of the research (pilot interview).
5. By providing *thick, rich descriptions* through contextualising data in the real-life practice of agile methodologies.
6. By consistently seeking *peer debriefing* of my supervision panel and frequently consulting with the chief investigator of this research.

5.5 Future Research

My exploratory study motivates further research about the role of conceptual modeling in practice. In addition to the proposed implications for theory and practice I discussed earlier, I see three significant areas for future research.

First, Representation Theory and the ontological framework of conceptual modeling have been extended primarily based on Bunge's ontology (Burton-Jones & Weber, 2014; Weber, 1997). As Wand and Weber (2002) point out, however, other ontologies and alternative philosophical assumptions can underpin other theoretical frameworks for conceptual modeling. Therefore, further research is needed to investigate how other frameworks could enrich our understanding of conceptual modeling. Investigating such frameworks could potentially lead to significant outcomes. For instance, it would be interesting to examine whether a *relational ontology* (Barad, 2014) rather than an *ontology of separateness* (Scott & Orlikowski, 2014) could provide a better conceptual fit (Burton-Jones, McLean, & Monod, 2014) for the new taxonomy of *ubiquitous open systems* (Lukyanenko & Parsons, 2013a).

Second, as complexity and uncertainty have been identified in my research as the most influential elements of context that impact conceptual modeling methods and scripts, further research could pursue how conceptual modeling grammars (and methods) could be adapted to support contexts where substantial uncertainty and complexity exist.

Third, because different data structures underpin different information systems, developing modeling methods that can link these non-matching structures, without a need to redesign the entire system, addresses a fundamental issue in practice. To address this issue, some new modeling techniques now adopt lean structures to maximise the interoperability of the structures that underlie information systems. Examining these lean structures and providing theoretical descriptions that could inform guidelines for practice based on the theories of conceptual modeling, enriches development of information systems. Examples of such alternative modeling grammars and methods are Data Vault (Jovanovic & Bojicic, 2012) and Pattern-Based Approaches (Giles & Ambler, 2012). Future research might investigate how simple modeling grammar in Data Vault, based on three notions of *hubs, links,* and *satellite*, represent domain semantics and improvise for interoperability of underlying structures of an information system. Similarly, examining Pattern-Based Approaches of modeling methods based on the ontological perspectives to conceptual modeling could provide deeper insight about guidelines for system development practices that are both scientifically based and meet the demands of the real-life projects.

BIBLIOGRAPHY

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (May 2003). *New directions on agile methods: A comparative analysis*. Paper presented at the 25th International Conference on Software Engineering, Portland, Oregon, United States.

Agarwal, R., Sinha, A. P., & Tanniru, M. (1996). Cognitive fit in requirements modeling: A study of object and process methodologies. *Journal of Management Information Systems, 13*(2), 137-162.

Ambler, S. (2005). Quality in an agile world. *Software Quality Professional, 7*(4), 34-40.

Ambler, S. (2014). Agile Modeling. 13 January 2014, from http://agilemodeling.com

Barad, K. (2014). Posthumanist performativity: Toward an understanding of how matter comes to matter. *Signs, 40*(1), 801-831.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. (2001). Manifesto for Agile Software Development. 13 January 2014, from http://agilemanifesto.org/

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly, 11*(3), 369-386.

Benediktsson, O., Dalcher, D., & Thorbergsson, H. (2006). Comparison of software development life cycles: A multiproject experiment. *IEE Proceedings-Software, 153*(3), 87-101.

Bodart, F., Patel, A., Sim, M., & Weber, R. (2001). Should optional properties be used in conceptual modelling? A theory and three empirical tests. *Information Systems Research, 12*(4), 384-405.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer, 35*(1), 64-69.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77-101.

Bunge, M. (1977). *Ontology I: The furniture of the world* (Vol. 3). Dordrecht, Holland; Boston, United States: D. Reidel Publishing Company.

Burton-Jones, A. (2012). Extending the Theoretical Program of Representation Theory *Information Systems Foundations Workshop*. Canberra: Australian National University.

Burton-Jones, A., McLean, E. R., & Monod, E. (2014). Theoretical perspectives in IS research: from variance and process to conceptual latitude and conceptual fit. *European Journal of Information Systems*, 1-16.

Burton-Jones, A., & Meso, P. N. (2006). Conceptualizing systems for understanding: An empirical test of decomposition principles in object-oriented analysis. *Information Systems Research, 17*(1), 38-60.

Burton-Jones, A., Wand, Y., & Weber, R. (2009). Guidelines for empirical evaluations of conceptual modeling grammars. *Journal of the Association for Information Systems, 10*(6), 495-532.

Burton-Jones, A., & Weber, R. (2014). Building conceptual modeling on the foundation of ontology *Computing handbook: Information systems and* *information technology* (3rd ed., Vol. 2, pp. 1-24). Boca Raton, FL, United States: CRC Press.

Clarke, R., Burton-Jones, A., & Weber, R. (2013). Improving the Semantics of Conceptual-Modeling Grammars: A New Perspective on an Old Problem.

Cockburn, A. (2002). *Agile software development*. Boston: Addison-Wesley.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research, 20*(3), 329-354.

Creswell, J. W. (2012). *Qualitative inquiry and research design: Choosing among five approaches*. Thousand Oaks: Sage publications.

Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory into Practice, 39*(3), 124-130.

Davis, G. B. (1982). Strategies for information requirements determination. *IBM Systems Journal, 21*(1), 4-30. doi: 10.1147/sj.211.0004

Denzin, N. K., & Lincoln, Y. S. (2000). *The handbook of qualitative research* (2 ed.). Thousand Oaks: Sage Publications.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management, 16*(4), 88-100.

Gemino, A., & Wand, Y. (2005). Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data & Knowledge Engineering, 55*(3), 301-326.

Giles, J., & Ambler, S. W. (2012). *The nimble elephant*. Westfield, NJ: Technics Publications.

Highsmith, J. (2002). *Agile software development ecosystems*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer, 34*(9), 120-127.

Hirschheim, R., Klein, H. K., & Lyytinen, K. (1995). *Information systems development and data modeling: Conceptual and philosophical foundations* (Vol. 9). Cambridge, New York: Cambridge University Press.

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). *Software quality and agile methods*. Paper presented at the International Computer Software and Applications Conference (COMPSAC 2004), Hong Kong.

Jovanovic, V., & Bojicic, I. (2012). *Conceptual Data Vault Model*. Paper presented at the Southern Association for Information Systems (SAIS), Atlanta, Georgia.

Kaplan, B., & Maxwell, J. A. (2005). Qualitative research methods for evaluating computer information systems *Evaluating the Organizational Impact of Healthcare Information Systems* (pp. 30-55). New York: Springer.

Khatri, V., Vessey, I., Ramesh, V., Clay, P., & Park, S.-J. (2006). Understanding conceptual schemas: Exploring the role of application and IS domain knowledge. *Information Systems Research, 17*(1), 81-99.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly, 23*(1), 67-93.

Kovitz, B. (2003). Hidden skills that support phased and agile requirements engineering. *Requirements Engineering, 8*(2), 135-141.

Krogstie, J., Lindland, O. I., & Sindre, G. (1995). Towards a deeper understanding of quality in requirements engineering *Seminal Contributions to Information Systems Engineering* (pp. 89-102). Chicago: Springer Berlin Heidelberg.

Kurtz, C. F., & Snowden, D. J. (2003). The new dynamics of strategy: Sense-making in a complex and complicated world. *IBM Systems Journal, 42*(3), 462-483.
Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer, 36*(6), 47-56.
Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data. *MIS Quarterly, 34*(1), 87-114.
Lincoln, Y. S., A., L. S., & Guba, E. G. (2011). Paradigmatic Controversies, Contradictions, and Emerging Confluences, Revisited *The Sage handbook of qualitative research* (4th ed., pp. 97-128). Thousand Oaks: Sage Publications.
Lindland, O. I., Sindre, G., & Solvberg, A. (1994). Understanding quality in conceptual modeling. *Software, IEEE, 11*(2), 42-49.
Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information Systems Management, 21*(3), 41-52.
Lukyanenko, R., & Parsons, J. (2011a). *Rethinking Data Quality as an Outcome of Conceptual Modeling Choices*. Paper presented at the 16th International Conference on Information Quality, University of South Australia, Australia.
Lukyanenko, R., & Parsons, J. (2011b). Unintended consequences of class-based ontological commitment *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6999, pp. 220-229).
Lukyanenko, R., & Parsons, J. (2013a). Is Traditional Conceptual Modeling Becoming Obsolete? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 61-73).
Lukyanenko, R., & Parsons, J. (2013b). Lightweight conceptual modeling for crowdsourcing *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8217, pp. 508-511).
Lyytinen, K. (1987). Two views of information modeling. *Information & Management, 12*(1), 9-19.
Marshall, M. N. (1996). Sampling for qualitative research. *Family Practice, 13*(6), 522-526.
Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Thousand Oaks: Sage Publications.
Milton, S. K., Rajapakse, J., & Weber, R. (2012). Ontological Clarity, Cognitive Engagement, and Conceptual Model Quality Evaluation: An Experimental Investigation. *Journal of the Association for Information Systems, 13*(9), 657-694.
Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions. *Data & Knowledge Engineering, 55*(3), 243-276.
Moody, D. L. (2009). The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *Software Engineering, IEEE Transactions on, 35*(6), 756-779.
Myers, M. D. (2013). *Qualitative research in business and management* (2nd ed.). London ; Thousand Oaks: Sage Publications.
Myers, M. D., & Avison, D. (1997). Qualitative research in information systems. *MIS Quarterly: Management Information Systems, 21*(2), 241-242.
Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization, 17*(1), 2-26.

Mylopoulos, J. (1992). Conceptual modeling and Telos *Conceptual modelling, databases and CASE: An integrated view of information systems development* (pp. 49–68). New York: Wiley.

Mylopoulos, J. (1998). Information Modeling in the Time of the Revolution. *Information systems, 23*(3), 127-155.

Nelson, H. J., Poels, G., Genero, M., & Piattini, M. (2012). A conceptual modeling quality framework. *Software Quality Journal, 20*(1), 201-228.

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM, 50*(3), 79-83.

Newman, I. (1998). *Qualitative-quantitative research methodology: Exploring the interactive continuum*. Southern Illinios University: SIU Press.

Offen, R. (2002). Domain understanding is the key to successful system development. *Requirements Engineering, 7*(3), 172-175.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research, 2*(1), 1-28.

Orlikowski, W. J., & Scott, S. V. (2008). Sociomateriality: Challenging the Separation of Technology, Work and Organization. *The Academy of Management Annals, 2*(1), 433-474.

Parsons, J. (2011). An experimental study of the effects of representing property precedence on the comprehension of conceptual schemas. *Journal of the Association for Information Systems, 12*(6), 401-422.

Parsons, J., & Wand, Y. (2000). Emancipating instances from the tyranny of classes in information modeling. *ACM Transactions on Database Systems (TODS), 25*(2), 228-268.

Parsons, J., & Wand, Y. (2008). Using cognitive principles to guide classification in information systems modeling. *MIS Q uarterly, 32*(4), 839-868.

Pederson, M. (2013). *A quantitative examination of critical success factors comparing agile and waterfall project management methodologies*. (Doctor of Philosophy), Capella University.

Phatak, O. (2012). Waterfall vs. agile model. http://www.buzzle.com/articles/waterfall-model-vs-agile.html. Retrieved 28 May, 2014

Potts, C. (1997). *Requirements models in context*. Paper presented at the 3rd IEEE International Symposium on Requirements Engineering, Annapolis, MD, USA.

Rajlich, V. (2006). Changing the paradigm of software engineering. *Communications of the ACM, 49*(8), 67-70.

Roussopoulos, N., & Karagiannis, D. (2009). Conceptual modeling: past, present and the continuum of the future *Conceptual Modeling: Foundations and Applications* (pp. 139-152). Chicago: Springer.

Royce, W. W. (1970). *Managing the development of large software systems*. Paper presented at the proceedings of IEEE WESCON, Los Angeles, California.

Sandberg, J., & Alvesson, M. (2011). Ways of constructing research questions: Gap-spotting or problematization? *Organization, 18*(1), 23-44.

Sarker, S., & Lee, A. S. (2006). Does the use of computer-based BPC tools contribute to redesign effectiveness? Insights from a hermeneutic study. *Engineering Management, IEEE Transactions on, 53*(1), 130-145.

Scott, S. V., & Orlikowski, W. J. (2014). Entanglements in practice: Performing anonymity through social media. *MIS Quarterly, 38*(3), 873-893.

Shanks, G., Nuredini, J., Moody, D., Tobin, D., & Weber, R. (2003, June 19--21, 2003). *Representing things and properties in conceptual modelling: An empirical evaluation*. Paper presented at the 11th European Conference on Information Systems, Naples, Italy.

Sharp, H., & Robinson, H. (2004). An ethnographic study of XP practice. *Empirical Software Engineering, 9*(4), 353-375.

Siau, K., & Rossi, M. (2011). Evaluation techniques for systems analysis and design modelling methods–a review and comparative analysis. *Information Systems Journal, 21*(3), 249-268.

Snowden, D. (2002). Complex acts of knowing: Paradox and descriptive self-awareness. *Journal of Knowledge Management, 6*(2), 100-111.

Sumrell, M. (2007). *From waterfall to agile-how does a qa team transition?* Paper presented at the Agile Conference (AGILE), 2007, Washington, DC.

Szalvay, V. (2004). An introduction to agile software development (pp. 1-9): Danube Technologies, Inc.

Thalheim, B. (2012). The science and art of conceptual modelling *Transactions on Large-Scale Data-and Knowledge-Centered Systems* (pp. 76-105). Chicago: Springer.

Thomas, J., & Harden, A. (2008). Methods for the thematic synthesis of qualitative research in systematic reviews. *BMC Medical Research Methodology, 8*(1), 45-55.

Thummadi, Lyytinen, K., & Berente, N. (2012). *Iterations in software development processes: A comparison of agile and waterfall software development projects*. Paper presented at the 7th Pre-ICIS International Research Workshop on Information Technology Project Management (IRWITPM 2012) (pp. 5-15).

Thummadi, Shiv, O., Berente, N., & Lyytinen, K. (2011). Enacted software development routines based on waterfall and agile software methods: Socio-technical event sequence study *In Service-Oriented Perspectives in Design Science Research* (pp. 207-222). Chicago: Springer.

Turk, D., Robert, F., & Rumpe, B. (2005). Assumptions underlying agile software-development processes. *Journal of Database Management, 16*(4), 62-87.

Urquhart, C. (2013). *Grounded theory for qualitative research: A practical guide*. Thousand Oaks, California: Sage Publications.

Walsham, G. (1993). *Interpreting information systems in organizations*. New York: John Wiley & Sons, Inc.

Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *European Journal of information systems, 4*(2), 74-81.

Walsham, G. (2006). Doing interpretive research. *European Journal of information systems, 15*(3), 320-330.

Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems, 15*(4), 285-304.

Wand, Y., & Weber, R. (1988). *An ontological analysis of some fundamental information systems concepts*. Paper presented at the In Proceedings of the Ninth International Conference on Information Systems (Vol. 1988, pp. 213-226).

Wand, Y., & Weber, R. (1990). An ontological model of an information system. *Software Engineering, IEEE, 16*(11), 1282-1292.

Wand, Y., & Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Information Systems Journal, 3*(4), 217-237.

Wand, Y., & Weber, R. (1995). On the deep structure of information systems. *Information Systems Journal, 5*(3), 203-223.

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling—a research agenda. *Information Systems Research, 13*(4), 363-376.

Weber, R. (1997). *Ontological Foundations of Information Systems*. Melbourne: Coopers & Lyberand.

Weber, R. (2003a). Conceptual modelling and ontology: Possibilities and pitfalls. *Journal of Database Management, 14*(3), 1-20.

Weber, R. (2003b). Still desperately seeking the IT artifact. *MIS Quarterly, 27*(2), 183-183.

Whetten, D. A. (1989). What constitutes a theoretical contribution? *Academy of Management Review, 14*(4), 490-495.

Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Thousand Oaks: Sage publications.

APPENDIX

A EXPLANATORY STATEMENT AND CONSENT FORM

MONASH University

EXPLANATORY STATEMENT

You are invited to take part in an exploratory study on *Quality of conceptual models arising in agile*.

Please read this Explanatory Statement in full before deciding whether or not to participate in this research. If you would like further information regarding any aspects of this project, you are encouraged to contact the researchers via the phone numbers or email addresses listed below this page.

What does the research involve?

In traditional methods of system development known as waterfall, the quality of information systems is determined by the fidelity between the formal representation of system specifications and the real world semantics. In this view, conceptual models are scripts that are formally representing system specifications via a modeling language and faithful representation of the underlying semantics of the real world is the measure of quality of conceptual models. High quality of the developed conceptual models is paramount to the quality of the ultimate information system. A theoretical framework that formalises this fidelity is called 'Theory of Representation' and an accompanying well-developed body of empirical experimentation based on this theory constructs our understanding of the notion of quality in Information Systems. However, the underlying assumptions of this perspective has been increasingly challenged by introduction of new types of systems such as open systems with heterogeneous and transient users, and methodologies such as agile.

The present study is an exploratory and qualitative research to examine the validity of underlying assumptions of Theory of Representation in an agile setting. We are interested in understanding the practice of modeling as it happens in real-life projects and verifying the implications of these assumptions for practice and theory. In this regard, while any practice of modeling is highly relevant to our research and therefore, all views are highly encouraged and valued, we have chosen agile as it explicitly challenges our theoretical assumptions. For instance, the significance of conceptual models in obtaining high quality information systems is either totally disregarded in agile view, or the modeling functionalities and definitions are completely transformed. By identifying the underlying principles of modeling in practice and examining the assumptions of our theory of representation, this research will potentially provide great insight in determining gaps between theory and practice. As part of this research, we are evaluating a set of high-level models that are developed in the course of an agile project, to see whether the predictions of Theory of Representation hold. We are also conducting interviews with expert data modelers and system architects to obtain their insight in relation to their practice of system development and their view of quality of scripts (models) arisen in an agile setting. This research will be published as a thesis towards a Master by Research degree. The thesis is equivalent to an extended article or a book chapter.

You are kindly invited to participate in this research. We believe that your experience in this area will provide invaluable insight into the present understanding of the phenomena. We are excited about this research and we believe that it would potentially make a great impact in extending both theory and practice.

Team of researchers

Prof. Ron Weber
Faculty of Information Technology
[redacted]

Naghmeh Sharikzadeh
Faculty of Information Technology
[redacted]

Dr. Caddie Gao
Faculty of Information Technology
[redacted]

Consenting to participate in the project and withdrawing from the research

Please kindly sign and return the consent form provided if you agree to take participate in this research. Participation in this research project is voluntary. If you do not wish to take part, you may withdraw even if you consent to participate and later change your mind. You are free to withdraw from this research project at any stage of this study. Information that you have contributed to the project can be withdrawn. However, withdrawal can only occur before your approval of the interview transcripts.

Possible benefits and risks to participants

There are no foreseeable direct benefits or risks to the participants. However, there would be a benefit to the community of practitioners and to the community in broad sense because of the insights this research will provide to the field and to the practice.

Confidentiality

Information that is collected during the interviews will only be used for the purpose of this study and great care will be taken to ensure that individuals' identities are protected. To ensure the collected data is treated confidential, the transcript of discussions will be presented to you for approval before it can be used in this research. Also, data will be aggregated anonymously and will contain no identifying information. Moreover, the collected information will be aggregated to ensure participants' identity remain anonymous when the data is published.

Storage of data

Data collected will be stored in accordance with Monash University regulations, kept on

University premises, in a locked filing cabinet for 5 years and only researchers of this centre can access it.

Use of data for other purposes

This data may be used towards further research as a PhD project by current researchers. All confidentiality measures that taken to protect anonymity and aggregation of de-identified data for the current project will be sustained if data is used towards a PhD research project.

Results

If you would like to be informed of the aggregate research finding, please contact Naghmeh Sharikzadeh on [redacted]. The findings are accessible no later than 6 months after data collection.

Complaints

Should you have any concerns or complaints about the conduct of the project, you are welcome to contact the Executive Officer, Monash University Human Research Ethics (MUHREC):

Executive Officer
Monash University Human Research Ethics Committee
(MUHREC)
Room 111, Building 3e
Research Office
Monash University VIC 3800

[redacted]

Thank you,

Professor Ron Weber

CONSENT FORM

Project: *Quality of conceptual models arising in agile*

Chief Investigator: Professor Ron Weber

I have been asked to take part in the Monash University research project specified above. I have read and understood the Explanatory Statement and I hereby consent to participate in this project.

I consent to the following:	Yes	No
Audio and/or video recording during the interview	☐	☐
Researchers in this project may use the data that I provide during this interview in future research.	☐	☐

Name of Participant ____

Participant Signature ____

B INTERVIEW PROTOCOL ADAPTED FROM CRESWELL (2012, P. 136)

Interview Protocol

Quality of Conceptual Models Arising in Agile

Time of Interview:

(Usually up to an hour unless confirmed with the participant for extended time prior to the interview or as the interview evolves)

Date:

Place:

(Preferably at a quiet conference room – recording facilities provided by myself)

Interviewee name:

Seek participant's consent to audio record the interview and obtain their signature on the Consent Form.

Provide a brief overview on the research, link it the Explanatory Statement and ask whether the participant requires any further information before commencing the interview.

Questions

Begin the interview with some general questions built upon the conversation already made with the participant in relation to the topic. The general questions should be along this line:

Main question: As my interest is in the conceptual modeling, I would like to know whether you used conceptual modeling in your practice?

If yes,

- Do you use/build conceptual models, data models or physical models?
- How do you use them in your practice?
- What techniques do you utilise to build them?
- Why do you think they are useful (strengths)?
- What are the shortcomings (limitations)?
- Do you think modeling has a future in system development?

If no,

- Why don't you use them?
- Does this mean that the purposes of conceptual modeling practices are no longer relevant? Or, are there alternative means to meet these objectives? (Traditionally, conceptual models are developed to attain four objectives: better domain understanding, basis for design, facilitating communication between stakeholders, and documentation of the original requirement of a system.)
- What information elicitation techniques do you use is system development and how do you evaluate the elicited requirements?
- What type of other modeling scripts do you use in your practice?
- What do you see as limitation and shortcomings of modeling that has led to the abandon of them in your practice?
- What are the strengths of your techniques?
- In your view, what are the limitations of your adopted practice?
- What is the future of system development in your view?

Note that the interviews should take no longer than an hour. Henceforth, as a courtesy matter, aim at completing the interview in 50 to 55 minutes to allow for a smooth closure of the interview, thanking the participant, re-assuring the confidentiality of responses and seeking for any potential referrals for further interview and their interest whether there need to be future interviews with the participant.