REFINTO: An Ontology-Based Requirements Engineering Framework for Business-IT Alignment in Financial Services

A thesis submitted to the University of Manchester for the
Degree of Doctor of Philosophy
in the Faculty of Humanities

2015

Emem Koffi Umoh

Manchester Business School

List of Contents

Word Count 47,992

List of Figures

List of Tables

Abstract

Business-IT alignment has been a top research topic for three decades now and consistently ranks high on CIO priorities and concerns. In spite of its seeming advantages, sustainable business-IT alignment remains elusive in practice. This can be attributed to the language and knowledge gaps which impede mutual understanding between business and IT stakeholders. It can also be attributed to the limitations imposed by approaching alignment solely from a strategic perspective. This thesis argues for an ontology-based framework that bridges the language and knowledge gaps through closer interaction between business and IT stakeholders throughout the software development and project management lifecycles, especially at the requirements engineering stage.

Attempts at achieving sustainable business-IT alignment predominantly focus on strategic alignment and have not been successful for various reasons. Firstly, driving down alignment initiatives to the operational and tactical levels is challenging. Secondly, it is difficult to operationalize the metrics used for evaluating alignment maturity at strategic levels. These limitations are less pronounced at the functional levels of an organization. It is at these levels that business strategies are executed and interaction between business and IT personnel is most frequent. The interaction between business and IT stakeholders in the execution of IT projects presents an opportunity that can be leveraged to drive alignment maturity.

The proposed framework is discussed in terms of its underpinning hypotheses, workflows, tool design and implementation, its use with a third party framework and tool. Antecedents to operational and tactical alignment such as quality, reuse, communication, learning, and shared understanding, are proposed as a practical means of achieving sustainable alignment maturity. The framework is applied to real-world, business-critical projects in a top global financial services organization and validated using descriptive statistical analysis and structural equation modelling techniques.

Contributions made through the study are highlighted. This includes the Alignment Forces Model which unifies the proposed framework and its support tool within software development and project management lifecycles. The Alignment Forces model and how it can be applied in practice is presented. Results of the quantitative data analyses indicate support for the arguments for the framework towards improving business-IT alignment, however with some limitations. Results also indicate support for the hypotheses for the antecedents to sustainable alignment maturity at lower organizational levels put forward. Finally, suggestions on furthering the study, addressing its limitations, and refining the framework and tool are articulated.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

My first and foremost thanks go to Dr. Pedro Sampaio whose guidance and support made this thesis possible. Over the last seven years he has ensured that I stay on track, reading through conference and journal papers, and providing insightful feedback. His passion and drive for producing high quality research is commendable. Special thanks also go to Dr. Babis Theodoulidis and Prof. Nikolay Mehandjiev whose constructive feedback and contributions during the bi-annual reviews have been invaluable. The content, quality, and extent of the research are a result of their insight and encouragement.

Thanks are also given to industry collaborators who gave up their time to be involved within the project. Organisations and individuals remain anonymous with maintained Non-Disclosure Agreements, however particular thanks are given to the numerous business sponsors, business function leads, domain experts, developers, project managers, business analysts and projects managers whose evaluation, feedback and contributions were invaluable to the project.

I would also like to thank my loyal and devoted group of friends for their continued support and enduring my conversations about the PhD on a regular basis for many years. I must mention a few – Ese, Segun, Gbemisola, Kabay, Pages family, Ademola, Edidiong, Kunle, and Henry. I must also acknowledge people who have mentored and encouraged me in ways they may never know – Agu Irukwu, Enoch Adeboye, TD Jakes, Jide Abolarin,Victor Ukwat, and Leke Sanusi.

Finally, my greatest thanks are given to my family. Their love, help, understanding and support through these years is far beyond any words and something for which I am eternally grateful.

The Author

The author of this thesis obtained a Master's degree in Information Systems (Merit) from the University of Manchester, UK in 2006 under the supervision of the current doctoral supervisory team. He also holds a Bachelors of Engineering degree in Computer Engineering (Second Class Upper) from the University of Uyo, Nigeria.

The author has held roles in business analysis, project management, application development, information risk management, IT audit, and cyber security. These roles have been in three Fortune 500 firms and one FTSE 100 firm across the financial services, supply chain management and logistics, energy, oil, and gas sectors.

The author's research interest and experience include post-graduate level research on service level management in the logistics and supply chain industry, application development, requirements engineering, business-IT alignment, ontological engineering, alignment evaluation methods, and knowledge engineering.

List of Acronyms

AE	Application Engineering
AFM	Alignment Forces Model
AI	Artificial Intelligence
API	Application Program Interface
BA	Business Analyst
BABOK	Business Analysis Body of Knowledge
BIA	Business IT Alignment
BPM	Business Process Management (BPM)
BSC	Balance Score Card
CFA	Component Factor Analysis
CMM	Capability Maturity Model
COBIT	Control Objectives for Information and Related Technology
DE	Domain Engineering
DSSE	Domain Specific Software Engineering
EA	Enterprise Architecture
FIBO	Financial Instrument Business Ontology
IEC	International Electro-technical Commission
IS	Information Systems
ISO	International Organisation for Standardisation
ITIL	Information Technology Infrastructure Library
ITSM	IT Service Management
ITGI	IT Governance Institute
JAD	Joint Application Design
JESS	Java Expert Shell
KBS	Knowledge-Based Systems
KE	Knowledge Engineering
MOF	Meta-Object Facility
MVC	Model-View-Controller
MVVM	Model-View-ViewModel
OE	Ontology Engineering
OEA	Ontology-enabled Architecture

ODD	Ontology-driven Development
OED	Ontology-enabled Development
OEA	Ontology-enabled Architecture
OKBC	Open Knowledge-Base Connectivity
OIS	Ontology Information Systems
OWL	Ontology Language
PCA	Principal Component Analysis
PMI	Project Management Institute
PM	Project Manager
PMLC	Project Management Lifecycle
PMBOK	Project Management Book of Knowledge
PRINCE2	PRojects IN Controlled Environments
RAD	Rapid Application Development
RBV	Resource-Based View
RDF(S)	Resource Description Framework (Schema)
RE	Requirements Engineering
RFST	REFINTO Framework Support Tool
ROI	Return on Investment
RM	Requirements Management
SAM	Strategic Alignment Model
SAMM	Strategic Alignment Maturity Model
SDLC	Software Development Life Cycle
SE	Software Engineering
SEM	Structural Equation Modelling
SPL	Software Product Lines
SPLE	Software Product Line Engineering
XML	Extensible Markup Language

Chapter 1 Introduction

1.1 Motivation

Organizations primarily invest in information technology (IT) to provide support to core business functions, improve efficiency and productivity, and gain competitive advantage. It is imperative that information technology processes, personnel, infrastructure, and strategy are geared towards achieving business goals. Divergence between business and IT goals, strategies, and priorities, can lead to delivery of IT artefacts that do not fully satisfy business needs. Reworking these artefacts to meet intended business needs often results in loss of time, resources, and competitive advantage. This can have an adverse impact on return on investment (ROI) in IT especially in dynamic and fast paced industries. On the contrary, organizations with aligned business and IT functions can gain competitive advantage, realise higher return on investment in IT, reduce time-to-market with products and services, and are agile to changes in the business environment. It is for these reasons that business-IT alignment (BIA) remains an interesting research subject.

Three decades on since the introduction of the strategic IT alignment concept (Henderson and Venkatraman, 1993), it remains elusive ((Coltman et al., 2015, Karpovsky and Galliers, 2015, Chan, 2002, Symons, 2005). Whilst the importance of the concept is acknowledged in both academia and practice, it remains abstract, fuzzy, hard to actualize in practice, and difficult to tell when attained (Regev et al., 2011, Silvius, 2007). Despite criticisms and cynicism about the concept, it remains a top issue for CIOs (Luftman et al., 2010) as organizations continue to seek out means to maximize ROI in IT. Whilst review of milestones attained in the research field indicate steady progress and sustained research interest, there are still gaps and shortcomings in extant literature (Chan and Reich, 2007). Firstly, in many cases, practical means to achieving sustainable alignment maturity is not proposed.

Secondly, the focus predominantly on strategic alignment which has been found to have limitations towards actualization of BIA leaves gaps with respect to exploring operational and tactical alignment. Thirdly, proposed antecedents, metrics, and models for measuring BIA maturity are better suited for strategic level alignment, leaving gaps in identification of antecedents, metrics, and measurement models for operational and tactical alignment. Fourthly, little is offered in terms of facilitating collaborative interaction between business and IT stakeholders in the performance of their functions routinely, with sustainable business-IT alignment as the goal.

Reference is made to *optimal* and *sustainable* business-IT alignment in this thesis. Optimal alignment takes into consideration the argument that rigid alignment between business and IT can be counterproductive by impeding organizational agility to environmental changes (Tallon and Pinosonneault, 2011, Benbya and McKelvey, 2006). This implies that getting the right type of alignment is imperative (Coltman et al., 2015). Sustainable alignment is alignment that is repeatable rather than fleeting and results when an organization's IT systems are developed in such a way that it adapts to changing business goals. It also means alignment which can be consistently and routinely attained in everyday operations of an organization's business and IT functions. This implies that business and IT functions in a sustainably aligned organization *co-evolve* (Vessey and Ward, 2013).

This thesis aims to make contributions to the furtherance of the understanding of business-IT alignment by proposing a means to attain sustainable BIA. To lay the foundation on which the research questions for this study are formulated, the main headings under which this thesis makes contributions are discussed briefly:

1. *Maturity ranking*. Given that BIA is desirable in organizations, it is imperative that organizations are able to objectively gauge the alignment between their business and IT functions. This leads to questions on how alignment maturity can be measured and means to improve alignment maturity. This continues to be the focus of extant literature mostly using the Strategic Alignment Maturity Model (SAMM) (Luftman, 2003, Luftman, 2000) and other maturity models based on it. There is a gap in extant literature for maturity measurement models

for operational and tactical alignment. This study attempts to close this gap by proposing a maturity measurement model using measures from SAMM and Balanced Score Card (BSC) that can be easily operationalized. The model is intended to gauge alignment at three stages of a software development project. These are during project scoping, planning, and requirements engineering activities (*pre-project*), during project execution (*intra-project*), and during the evaluation after execution (*post-project*).

2. *Metrics*. Defining appropriate metrics that can be easily operationalized towards maturity ranking is necessary for effective measurement of alignment maturity. As with maturity ranking, there is a gap in extant literature on metrics for gauging operational and tactical alignment maturity. Whereas some of the six SAMM measures for gauging strategic alignment may be appropriate for gauging BIA at strategic levels not all can be operationalized for gauging alignment maturity at tactical and operational levels. The identification of other metrics not in SAMM but can be relevant to measuring alignment maturity at operational and tactical levels can contribute to enriching our knowledge of BIA and towards practical means of attaining sustainable BIA. This study attempts to address this gap in extant literature by proposing and validating metrics for alignment at operational and tactical alignment by applying them to projects in the real world.

3. *Levels of alignment*. Alignment can be observed at the three levels of an organization namely strategic, operational, and tactical. Extant literature on BIA focus predominantly on alignment models, methods, and metrics targeted at the strategic level. These models tend to follow a top-down approach with business strategy as the primary driver. Following this approach can lead to the assumption that BIA is a static model that measures strategic fitness and functional integration between organizational units rather than a process for controlling the organization for the implementation of goals (Henderson and Venkatraman, 1993). The top-down approach has shortcomings such as difficulty in translating strategic level alignment objectives to tactical operational levels actions and empirically measuring BIA maturity at this level.

In this study, BIA is approached from the operational and tactical levels by leveraging the interactions between IT and business functions during project implementation, technology choices, resource allocation, requirements, IT delivery capabilities, and IT governance. This is proposed as a means to attaining sustainable BIA.

4. *Antecedents*. Closely related to the maturity ranking and metrics is antecedents to BIA, which refer to factors that can enable or inhibit BIA maturity. Identification and validation of antecedents to BIA at strategic levels is investigated in extant literature. The identified antecedents tend to be better suited to strategic level (Reich and Benbassat, 2000, Chan et al., 2006, Yayla and Hu, 2009). The validation of these antecedents is performed predominantly by using qualitative surveys targeted at management level stakeholders with no practical applications of the antecedents in real life projects. This study contributes to addressing this gap through the identification and validation of antecedents to operational and tactical alignment through practical application to projects in the financial services domain.

5. *Point of intervention*. Closely related to the level of alignment, maturity models, and antecedents to BIA, is the determination of the point of intervention to advance BIA maturity. This can either be from the strategic perspective (top-down) or functional (operational and tactical) perspective. The perspective taken can have an influence on research direction on the BIA concept. For example, some definitions of BIA seem to be influenced by the point of intervention. A top-down approach (macro-foundational) to BIA, the predominant approach, starts at the strategic level whereas a bottom-up approach (micro-foundational) starts at the operational and tactical levels. An approach that considers both a top-down and bottom-up perspectives referred to as the co-evolutionary approach has been proposed (Benbya and McKelvey, 2006). This study explores an approach starting at the operational and tactical levels on individual project basis on the assumption that micro-foundational alignment attained can aggregate to macro-alignment (strategic alignment) over a period of time.

6. *Means of Intervention*. An interesting perspective to view the BIA concept from is the means a BIA maturity uplift effort is driven. Three approaches with potential to facilitating sustainable alignment proposed in extant literature are architecture, communication, and governance (Chen, 2008). BIA via architecture involves using business analysis and architecture analysis to align elements into three layers of business systems namely, business model, business architecture, and IT architecture. BIA via governance mainly involves service and business performance management. The objective being to ensure that organizational plans and actions arising from business and IT domains are properly integrated. BIA via communication involves bridging 'cultural gaps' and 'language gaps' between business and IT stakeholders. This study approaches business-IT alignment mainly through communication with aspects of architecture and governance also used.

1.2 Research Question Statement

Based on the gaps in extant literature on business-IT alignment this study aims to contribute to addressing, the key research questions are articulated in this section. The main research question explored in this study is:

> *Would an ontology-based requirements engineering framework and tool-supported approach applied to agile/RAD applications projects implemented at tactical and operational levels cumulatively contribute to improved business and IT alignment at these levels?*

To adequately address this research question, sub research questions explored in the study are:

1. Can this framework be made to support intelligent and on-demand reasoning and decision making during the requirements engineering process, during project implementation, and during evaluation after project implementation?

2. Would the framework facilitate a more structured and rigorous requirements engineering process than an ad hoc approach in agile/RAD projects without encumbering the process with the demands of the waterfall development methodology such as delaying projects till formal requirements documents are signed off?
3. By what metrics and means can BIA be measured at tactical and operational levels and can these be evaluated through real life projects using the proposed framework? Based on this, can the framework, antecedents to BIA and metrics for BIA maturity be integrated to software development and project management lifecycle?
4. What are the antecedents to BIA and forces that influence BIA at the tactical and operational levels?

The main research questions and the sub research questions are articulated further in the research design discussion in chapter four.

1.3 Research Objectives

The objective of this study is to contribute to the understanding of the BIA concept by proposing a means to actualize the concept in practice. The aim is to propose and validate a framework that is useful for achieving and sustaining BIA at tactical and operational levels. The use of ontology-based requirements engineering framework and tool starting at the requirements engineering stage of agile/RAD projects, during development, and post-implementation is used to contribute to closing the knowledge and language gaps that can exist between business and IT stakeholders in highly technical industries like the financial services.

It is also an objective of this study to propose a means of introducing structure and rigour to the requirements engineering process especially in agile/RAD projects without negatively impacting on time-to-market expectations. This is based on the argument that BIA maturity improvements obtained in individual projects can translate to BIA maturity improvements at operational and tactical levels and cumulatively, at strategic levels.

This study aims to identify metrics for measuring BIA maturity suited to tactical and operational levels, relevant to evaluating requirements quality, implementation processes, and gauging the impact of processes followed on project outcomes. This is also expected to contribute to addressing the gaps in extant literature on metrics for operational and tactical alignment maturity and validated in this study.

A significant contribution this study aims to make is to propose a model that maps software development life cycle (SDLC) to project management life cycle (PMLC) and integrates the identified antecedents, metrics, and ontology-driven iterative processes into a holistic model specifically focused on business-IT alignment. This is expected to contribute to addressing a gap in extant literature for practical means of actualizing business-IT alignment. The proposed *Alignment Forces Model* is envisaged to be useful to practitioners and academics seeking a comprehensive model for driving sustainable BIA maturity.

1.4 Research Method

The research method followed in this research is design research (DR). Although case study and action research (AR) can be followed for a research of this nature, design research is adopted because it is considered as a better fit for the study. Furthermore, design research has been followed successfully in similar doctoral studies (Al Balushi, 2008, Edge, 2010, Al Balushi et al., 2013). Design research is defined as:

"The analysis of the use and performance of designed artefacts to understand, explain and very frequently to improve on the behaviour of aspects of Information Systems. Such artefacts include – but certainly not limited to – algorithms (e.g. information retrieval), human/computer interfaces and system design methodologies or languages" (Blessing and Chakrabarti, 2009, Vaishnavi and Kuechler, 2007)

DR begins with the awareness of the problem and the suggestions for solving the problem are abductively drawn from existing knowledge/theory base of the problem domain before the implementation of an artefact based on the suggested solution is undertaken. Partial or fully successful implementations of the solutions are then evaluated. The suggestion, implementation and evaluations steps of the design research can be iterative.

The steps in a DR study are now considered with focus on how this study maps to these steps as depicted in Figure 1.1:

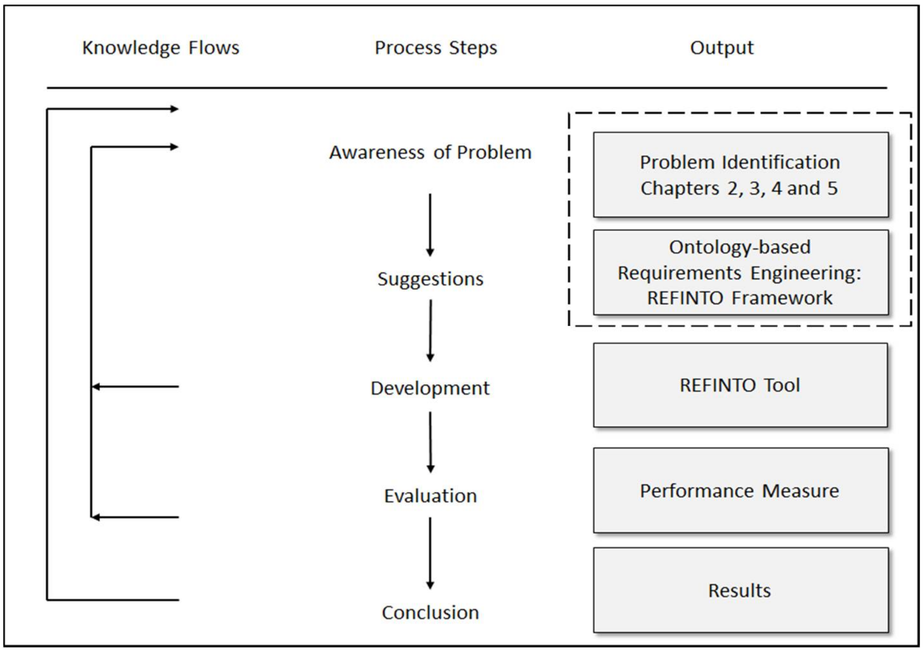

Figure 1.1 The General Methodology of Design Research

1. Awareness of Problem

The impact of misalignment between business and IT can be severe. It can range from building IT artefacts that are not fit for purpose, loss of resources, loss of reputation, to the loss of competitive advantage (Ralha and Gostinski, 2008). The first part of the thesis provides an overview of issues related to BIA, requirements engineering, and knowledge-based requirements engineering approaches to BIA. The second part of the thesis describes the proposed ontology-based requirements engineering framework, its implementation, and application to agile/RAD projects with the aim to assessing its impact on BIA maturity at tactical and operational levels.

2. Suggestion

The second part of the thesis focuses on the proposed ontology-based requirements engineering framework and tool. The suggestion made is to drive alignment maturity in a number of ways. Firstly, leveraging the close interaction and collaboration between business and IT stakeholders, secondly, by bridging the knowledge and language gaps between business and IT stakeholders, and thirdly, by facilitating a structured and rigorous approach to agile/RAD project requirements engineering and management with business-IT alignment as the goal. The REFINTO framework is proposed as a means to guide requirements elicitation, reuse, and refinement during the requirements engineering stage, requirements to artefact matching during implementation, and evaluation post-implementation. This is expected to contribute to improving business-IT alignment maturity.

3. Development

The development activities performed in the course of this study include:

- The design and implementation of the REFINTO framework and tool for requirement elicitation, reuse, refinement, persistence, data collection, and statistical analysis, in Java, JESS, Microsoft.net C#, and Protégé.
- Building of a repository of past requirements, constraints, and corresponding assets (libraries, classes, methods, etc.) related to the requirements.
- Development of 15 domain ontologies for recurring business functions within financial services such as reconciliation, data sourcing, reporting, etc.

- Identification and operationalization of metrics suited for gauging BIA at operational and tactical levels.

4. Evaluation

The main objective of the evaluation activity in this study is two-fold. Firstly, to validate the argument that structured and rigorous approach to requirements engineering and subsequent SDLC activities can contribute to producing high quality requirements and improved BIA maturity. Secondly, to validate the hypotheses proposed in the study as antecedents to operational and tactical BIA and to validate the metrics identified for gauging operational and tactical alignment on a per-project basis.

To achieve this, three stages of projects are evaluated. These are pre-project (the requirements engineering stage), the project implementation stage (intra-project) and the project closure stage (post-project). An instrument for evaluating BIA at these stages is developed. The instrument consists of questions for each of these stages based on metrics that have been identified from theory and practice, an approach followed in extant literature (Chen, 2010, Khaiata and Zualkernan, 2009, Guiterrez et al., 2006). Four portfolios of projects are evaluated, two using the REFINTO framework (one set of projects with tool support and another set of projects without), one without any framework, and one with a third-party framework as control.

5. Theorizing

The hypotheses on antecedents to operational and tactical BIA are put forward in this study. Data collection and analysis is performed to empirically validate the hypotheses. Based on the hypotheses and the argument for leveraging RE, SDLC, and PMLC for driving sustainable BIA, the *Alignment Forces Model* is proposed. The iterative process of the model can be integrated with software development and project management lifecycles processes to compliment agile development methodologies.

1.5 Research Context and Assumptions

This thesis focuses on achieving sustainable BIA maturity starting at operational and tactical levels through structured and rigorous requirements engineering and management processes for agile/RAD projects in a fast moving and rapidly evolving industry - the financial services domain. The framework is validated in the context of this domain to test its ability to facilitate closer involvement, interaction, and collaboration between business and IT stakeholders during conceptualization, planning, and execution of software development projects in an industry with high requirements volatility driven by competition and regulatory demands.

The nature of iterative software development methodologies and agile/RAD projects makes them particularly interesting in the context of this thesis. The distinguishing features of agile/RAD projects include changing and incomplete requirements. While these features allow development to begin before requirements are complete or fully understood and requirements change at any point in the development lifecycle, these projects are susceptible to business-IT misalignment issues. Whereas, iterative development methodologies are widely adopted in the financial services industry, it is not in its purest form. This is due to compliance, audit, and regulatory stipulations that require evidence of planning, execution, testing, and change management typically associated with waterfall development methodologies, especially for financial transaction and reporting applications. The focus on agile/RAD projects and iterative software development methodologies in this thesis was because of its relevance to business-IT alignment. Furthermore, the projects for which data is collected and analysed in this study were agile/RAD projects.

The financial services domain is interesting for various reasons. Firstly, there are factors within the domain that make it more susceptible to misalignment between business and IT. The large vocabulary of specialized business terms, national, regional, and international regulations, bespoke and proprietary product offerings can exacerbate the language and knowledge gaps between business and IT stakeholders. Secondly, the rapidly changing, time-sensitive business needs, and regulatory requirements make the need for agility to respond to these changes imperative and heighten the likelihood of misalignment. Thirdly, the vast number of processes, services, and assets built up over time presents an opportunity for reuse and service orientation.

The focus on financial services notwithstanding, the research output of this study is expected to be relevant to other domains. The limitations of the framework and tool and opportunities for further research on the subject are highlighted.

The assumptions made in this are:

- Alignment maturity attained on a per-project basis or for a portfolio of projects when aggregated can be extrapolated to give an indication of operational and tactical alignment maturity.
- Alignment maturity can be improved when the right structures and rigorous processes are in place for requirements engineering and management in an organization.
- Although an agile/RAD (iterative) development methodology is assumed, it does not limit the applicability of the framework, tool, metrics, and models proposed to iterative methodologies.
- The projects used to validate the proposed framework, tool, metrics, and models are generally those that run from a few weeks to a couple of months. This does not limit applicability to projects of this length. Multi-year projects can also be supported.

1.6 Research Contributions

The main contributions of this research are:

1. Proposal of a practical means of attaining sustainable business-IT alignment through a framework that improves interaction and collaboration between business and IT stakeholders by bridging language and knowledge gaps that

exists due to the large vocabulary and specialist knowledge, which can hinder effective requirements engineering, project execution processes, and subsequently, the delivery of IT artefacts. This framework when applied at the requirements engineering stage of application development projects ensures that project participants are empowered with a means to collaboratively elicit, reuse, refine, and persist high quality requirements. This introduces structure and rigor into a process that can otherwise be ad hoc and chaotic.

2. Identification of metrics and measurement models that can be used to measure requirements quality and requirements implementation processes. It can also be used to gauge the level of business-IT alignment attained in projects. The impact of this contribution is that metrics that can be operationalized in a daily operation of organizations seeking to improve business-IT alignment maturity are identified.

3. Proposal of eight antecedents to operational and tactical level business-IT alignment validated through a combination of qualitative data collection processes and quantitative analysis of data from real world projects in the domain of study.

4. Proposal of an ontology-based requirements engineering approach to agile/RAD projects (Umoh et al., 2011, Umoh et al., 2012). This involves the use of domain and business knowledge codified in domain ontology to facilitate mutual understanding between business and IT during the requirements elicitation process. This leads to streamlining the process, curbing ambiguity, bridging language barrier, and ensuring business and IT have a mutual understanding of business requirements.

5. The tool developed to support the proposed framework is useful for supporting knowledge-based requirements elicitation, reuse, refinement, and management. The tool utilizes domain ontologies for search and reuse of closely related past requirements, constraints, and related artefacts from historic projects. This

leads to production of high quality requirements, reduced elicitation time, and enhance organizational and personal learning, and knowledge sharing.

6. The alignment forces model (AFM) which unifies the REFINTO framework and the identified antecedents to business-IT alignment at tactical and operational levels, can be used in real-life projects following agile and iterative development methodologies.

1.7 Structure of Thesis

The thesis is organized in eight chapters split into two parts. Part One (Overview), summarizes relevant concepts of BIA, requirements engineering, knowledge-based approaches to requirements engineering, software development, and project management lifecycles from the perspective of business-IT alignment:

- Chapter 2. Business-IT Alignment and Requirements Engineering
  This chapter covers the literature review of the key concepts on business-IT alignment and requirements engineering in the context of business-IT alignment. Working definitions, relevance of BIA as a research subject, approaches, BIA measurement models, and theoretical underpinning of BIA in extant literature are reviewed. The link between requirements engineering and business-IT alignment is explored. It is argued that sustainable BIA can be attained by leveraging the interaction and collaboration of business and IT stakeholders during requirements engineering and project execution activities involving elicitation, analysis, and specification of the information about the application domain or problem domain the software is intended to function in.

- Chapter 3. Knowledge-based Approach to Requirements Engineering and Business-IT Alignment
  The process of capturing requirements for reuse involves knowledge management. This chapter makes an argument for a knowledge-based

requirements engineering framework specifically designed for business-IT alignment at the tactical and operational levels of an organization. The concepts and activities of domain engineering, knowledge engineering, and ontological engineering are reviewed with respect to business-IT alignment.

- Chapter 4. Research Design

  This chapter covers the research method, design, and strategy adopted in this study. The design research method introduced in chapter one is explored further and its choice for this study is articulated. This study is a deductive and mainly exploratory research. A post-positivist and pragmatist stance is taken. The data collection method is through role-based and stage-differentiated surveys. The data analysis method adopted is mainly quantitative. Statistical analysis is performed on the data collected and structural equation modelling (SEM) is used to validate the hypotheses and the correlation between the factors/constructs in the study.

Part Two (Contribution), comprises:

- Chapter 5. REFINTO Framework and Support Tool – Concepts, Design and Development

  This chapter concentrates on the RFFINTO framework and support tool. The components of the framework, role-based and process-based workflows, and hypotheses underpinning the framework, architecture and design principles of its support tool are discussed. A comparative analysis of the framework with alternative frameworks in relation to business-IT alignment is presented. The ontology-based requirement-artefact matching process for elicitation and refinement of requirements is explained. The measurement and evaluation mechanism of the REFINTO framework is also presented.

- Chapter 6. Data Collection and Analysis

  This chapter focuses on the data collection and analyses performed to validate the hypotheses and the effect structured and rigorous requirements engineering and management processes have on BIA maturity. The strategy adopted for data collection and the instrument designed for data collection are discussed.

- Chapter 7. Interpretation of Results, Evaluation and Synthesis of Theories
  This chapter concentrates on the interpretation of the results of the data analysis performed in chapter six and if this supports the hypotheses or otherwise. Major findings from the data collection and analysis are highlighted. The limitations of the framework, tool, and processes proposed and developed are also highlighted. The alignment forces model is presented in this chapter.

- Chapter 8. Contributions, Conclusions and Future Work
  This chapter focuses on the contributions this study makes to the BIA research. The challenges encountered during the research and suggestions to improve the study are articulated. A summary of the research findings against the research questions and objectives are discussed to highlight the contributions this study makes to the understanding and actualization of business-IT alignment.

Chapter 2 Business-IT Alignment and Requirements Engineering

2.1 Introduction

Business-IT alignment (BIA) has attracted considerable research interest for about three decades. Although significant milestones have been reached in the BIA research, it remains elusive in practice. This can be attributed to the focus on strategic level business-IT alignment. This thesis argues for approaching business-IT alignment from functional (operational and tactical) levels, at which interaction between business and IT stakeholders is most intense as a means to achieve sustainable BIA. This is premised on the assumption that the interaction of business and IT stakeholders, processes, and functions during software development and project management lifecycles, especially at the requirements engineering (RE) stage, can be leveraged to attain optimal and sustainable BIA at operational and tactical levels of an organization.

The execution of business plans and strategies often require implementation of IT solutions such as in-housed, third-party developed, or customized off-the-shelf software applications. These activities involve close interaction between business and IT stakeholders who may have different levels of business domain knowledge and understanding. This interaction starts at the requirements gathering stage, continues during the implementation stage, and up to deployment. The requirements gathering stage is difficult to get right and critical to project outcomes. The effectiveness of communication between business and IT stakeholders at this early stage and throughout the project lifecycle can contribute to the success or failure of the project. To this end, a common language between business and IT stakeholders for mutual understanding and communication of business requirements and priorities is imperative. Furthermore, processes, procedures, and frameworks applicable to appropriate software development and project management lifecycles with sustainable BIA as the end goal is desirable.

Close interaction between business and IT is necessary for ensuring timely delivery of projects and meeting business needs. It can therefore be argued that there is a link between requirements engineering (RE) and BIA. This link is explored in this chapter. To achieve this, RE activities, requirements management (RM), and RE/RM tools are explored from the perspective of relevance to BIA. The argument made is that RE activities can be leveraged to achieve sustainable BIA. This is premised on the argument that user involvement, the use of appropriate development methodology, and RE practices are critical to software project outcomes (Doherty et al., 2012, McLeod and MacDonell, 2011, Dvir et al., 2003) and by extension business value and return on investment (ROI) in IT (Byrd et al., 2006, Avison et al., 2004, Kashanchi and Toland, 2006).

The objective of this chapter is to review the progress made so far in the BIA research, highlight gaps in literature that this study attempts to address, and form the building blocks on which the contributions made through this study are based. Definitions, approaches, alignment levels (strategic, tactical, and operational), antecedents, maturity models, metrics, measurement methods, theories applied to BIA, and RE concepts relevant to BIA are also reviewed. The chapter ends with a critical analysis of the impact requirements engineering has on BIA and lays the foundation for the next chapter, which considers knowledge-based requirements engineering approaches to BIA.

2.2 Working Definitions

There are numerous definitions for BIA using a variety of terms to describe the concept. This indicates some misunderstanding of the concept and the goals it aims to achieve (Maes et al., 2000). A selection of terms used in defining business-IT alignment is presented in in Table 2.1. It can be argued that BIA definitions are influenced by the perspective from which it is considered, such as a top-down process, a bottom-up process, and sphere of experience – academic, industry or both (Umoh and Sampaio, 2014). A selection of BIA definitions is presented in Table 2.2. The definition adopted in this thesis is the one put forward by Benbya and Mckelvey (Benbya and McKelvey, 2006). This definition is adopted because it captures the dynamic nature of BIA.

Table 2.1: Equivalent Terms for Business IT Alignment

Synonym	References
Bridge	(Ciborra, 1993, Strnadl, 2005)
Coordination	(Lederer and Mendelow, 1986)
Fit	(Bergeron et al., 2004, Venkatraman, 1989)
Fusion	(Smaczny, 2001)
Harmony	(McKeen and Smith, 2003, Woolfe, 1993, Luftman, 2000)
Integration	(Weill and Broadbent, 1998)
Linkage	(Henderson and Venkatraman, 1993, Reich and Benbassat, 1996)
Match	(Palmer and Markus, 2000, Motjolopane and Brown, 2004, Cumps et al., 2006b)
Synchronization	(Luftman and Rajkumark, 2007)

The working definition adopted for requirements in this thesis is the IEEE 610.12-1990 standard (IEEE, 1990), definition that a requirement is:

"*A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents*".

The requirements engineering definition adopted in this thesis is the definition by Nuseibeh and Easterbrook (Nuseibeh and Easterbrook, 2000), which highlights stakeholder involvement in the process. They define requirements engineering thus:

"*Software systems requirements engineering, (RE) is the process of discovering that purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation*".

Table 2.2: Definitions of Business IT Alignment

Definition	Perspective/Focus
The degree to which IT applications, infrastructure and organization enable and support the business strategy and processes, as well as the process to realize this (Silvius et al., 2009)	Strategic/Top-Down/Academic
The degree to which the information technology mission, objectives and plans support and are supported by the business mission, objectives and plans (Chan, 2002)	Strategic/Top-Down/Academic.
Strategic Alignment of IT occurs when harmony exists between firm’s goals and activities and the implemented information systems (McKeen and Smith, 2003)	Strategic/Top-own/Practitioner.
The degree of fit and integration among business strategies, IT strategies, business infrastructures, and IT infrastructures (Henderson and Venkatraman, 1993)	Strategic/Top-down/Practitioner
Applying IT in an appropriate and timely way, which is in harmony with business strategies, goals, and needs (Luftman, 2000)	Strategic/Top-Down/Academic.
The degree to which the mission, objectives, and plans contained in the business strategy are shared and supported by the IT Strategy (Reich and Benbassat, 1996)	Strategic/Top-Down/Academic
Coordination of a firm’s IT strategy with its business strategy (DeLisi, 2005)	Strategic/Top-Down/Practitioner
The continuous process, involving management and design sub-processes, of consciously and coherently interrelating all components of the business–IT relationship in order to contribute to the organisation’s performance over time (Maes et al., 2000)	Strategic/Top-Down/Academic
The process through which business people and IT delivery organisations collaborate to create an environment in which investment in IT and delivery of IT services reflect business priorities, whether IT services are sourced internally or externally; and in which business priorities are influenced by understanding of IT capabilities and limitations (Macehiter and Ward-Dutton, 2005)	Strategic/Top-Down/Practitioner
A continuous coevolutionary process that reconciles top-down ‘rational designs’ and bottom-up ‘emergent processes’ of consciously and coherently interrelating all components of Business/IS relationships at three levels of analysis (strategic, operational and individual) in order to contribute to an organization’s performance over time (Benbya and McKelvey, 2006).	Strategic/Coevolutionary/Academic

2.3 Importance of Business-IT Alignment

The degree of business-IT alignment maturity in an organization can have tangible and intangible impact on the organization. There is support in extant literature for the claims that business-IT alignment gives an organization competitive advantage, improved performance, improved productivity, and agility to react to business environment changes (Yayla and Hu, 2012, Sabherwal and Chan, 2001). There is, however, a school of thought that questions these claims, especially with respect to strategic alignment (Tallon and Kraemer, 2003, Jarvenpaa and Ives, 1994, Papp, 1999). It can be argued that if alignment is beneficial to an organization then misalignment can have undesirable consequences The tangible and intangible impact of BIA are summarized in Table 2.3.

BIA became a core managerial concern due to organizations' increasing dependence on information systems (Silvius, 2007). This is further complicated by the mismatch between the speed of change required to drive strategic initiatives and meet regulatory demands. BIA continues to attract interest in industry due to sustained focus on maximizing business value from IT investments. Consequently, BIA retains its top ranking on CIOs concerns (Huang and Hu, 2007).

Table 2.3: Categorization of Impacts of BIA

	Tangible	Intangible
Alignment	Efficient use of resources Improved Performance Increased return on investment in IT Improved time-to-market	Competitive Advantage Goodwill Enhanced reputation Agility to respond to market opportunities
Misalignment	Loss of resources	Loss of reputation/Reputational Damage Inability to respond quickly to business opportunities Impaired competitive advantage

2.4 Approaches to Business-IT Alignment

Numerous approaches are proposed for achieving BIA maturity in extant literature. These include architecture, governance and communication (Chen, 2010, Chen, 2008) which show promise towards sustainable BIA. Other strategies such as parallelism, profiling, and leadership synergy tend to lead to short term alignment (Samanta, 2007). Typically, an alignment maturity uplift effort involves a combination of approaches if optimal and sustainable alignment maturity is to be attained (Umoh et al., 2011). In this section, the architecture, governance, and communication approaches to business-IT alignment are reviewed. This lays the foundation for the REFINTO framework, which mainly follows the communication approach with some aspects of the architecture and governance approaches.

2.4.1 Alignment via Architecture

The Architecture-based approach employs business analysis, business modelling, business architecture, IT architecture, cost benefit analysis, and architectural decision patterns to drive alignment. An example is Service Oriented Modelling and Architecture (SOMA) (Abdi and Dominic, 2010, Dodani, 2007). The architecture approach to BIA hinges on enterprise architecture (EA). The EA discipline aims to achieve BIA through architectural oversight and governance (Bradley et al., 2012).

EA is a framework for managing and aligning an organization’s asset, people operations, and projects as part of its operational routine (Wang et al., 2008, Abdi and Dominic, 2010). Its four-dimensional model (business, information, application and technical architecture) are aimed at facilitating analysis, technology choice, and decision-making to enable and support business strategies (Banarjee and Aziz, 2007). The four dimensions of EA are described in Table 2.4. The relationship between these dimensions of EA and BIA is then discussed briefly

Table 2.4: EA Dimensions (Banarjee and Aziz, 2007)

EA Dimension	Description
Business Architecture	Defines the business strategy, governance, organization, and key business processes
Applications Architecture	Provides a blueprint of the individual application systems to be deployed, their interactions and relationship with the core business processes of the organization
Information Architecture	Describes the structure of an organization's logical and physical data assets and data management resources
Technology Architecture	Describes the software infrastructure that supports the deployment of core, mission-critical applications. This type of software is sometimes called the "middleware"

Whereas business architecture relates to the business specifically, the other three dimensions, grouped and referred to as IT Architecture, relate to design and implementation of technology artefacts. To achieve sustainable business-IT alignment through architecture, it is essential to move the focus of enterprise architecture from a technology-centric perspective to a business-centric perspective.

Business architecture is concerned with aligning the development and maintenance of enterprise capabilities with business strategy and plans (Aier, 2009, Banarjee and Aziz, 2007). This is also the objective of strategic alignment as captured in the Strategic Alignment Model (SAM) (Henderson and Venkatraman, 1993) which highlights the linkage and integration of these constructs for optimal alignment between business and IT.

Information architecture relates to the structuring of an organization's logical and physical data assets and data management resources is considered. If the strategy for the structuring is driven purely by technical considerations and the business needs are overlooked, it will invariably lead to misalignment issues and inability to meet business needs quickly and efficiently. This can be further exacerbated by the lack of a strategic roadmap for information architecture in an organization, which can hinder the exploitation of an organization's data assets for business intelligence and maximizing competitive advantage.

Application architecture is mainly software-related and can be leveraged to attain sustainable BIA by facilitating the delivery of application solutions to the business in a number of ways. Firstly, the quality of elicited requirements has direct impact on the architectural designs and subsequent implementation of software artefacts. Secondly, requirements patterns and reuse imply that artefacts that are developed to meet recurring requirements can also be reused.

Technical architecture focuses on technology choices, which should be driven by current and anticipated business needs. Technology choices can inhibit or enable future business strategy like new product offerings. To facilitate sustainable business-IT alignment maturity, it is necessary that future business needs be taken into consideration when IT investments are made.

Service Orientation Architecture (SOA), an architectural paradigm for reuse, has potentials for BIA alignment maturity through well-defined services aligned to business, and developed with established design principles, frameworks, patterns and methods (Banarjee and Aziz, 2007). SOA follows two main approaches, a process-driven (top-down) approach and an application-driven (bottom-up). The process-driven approach works by identifying and categorizing services that can be implemented in such a way to enable choreographing them to provide scalable and flexible on-demand solutions. The application-driven approach works by identifying areas of loose coupling and reusability before the development of core services.

2.4.2 Alignment via Governance

Alignment through governance aims to ensure that organizational plans and actions originating from both the business and the information technology domain are properly integrated. It also includes value delivery, resource management, risk management, and performance measurement, and organizational transformation

(Gronau and Rohloff, 2007). It is argued that one of the goals of IT governance is achieving better alignment between business and IT (Van Grembergen et al., 2003).

Review of extant literature indicates support for alignment via governance. Firstly, there is strong support for the argument that BIA maturity is higher in organizations that apply a mix of mature IT governance practices (Haes and Van Grembergen, 2009a, Haes and Van Grembergen, 2009b). Secondly, it indicates support for the argument that governance through executive support influences business-IT alignment (Schlosser et al., 2015). This can be structurally through IT representation at board level and behaviourally through senior management support for alignment maturity uplift activities at strategic and operational levels (Beimborn et al., 2009).

The governance approach to BIA has two main streams – Business Process Management (BPM) and IT Service Management (ITSM). BPM is cited as a methodology that can be used to systematically attain and sustain strategic alignment by facilitating the creation of strategic goals and supporting the performance management of those goals (Ariyachandra and Frolick, 2008, Frolick and Ariyachandra, 2006). A well-known BPM framework that has been adapted for use in addressing BIA is Balanced Score Card (BSC) (Kaplan and Norton, 1996, Kaplan and Norton, 1992) is widely adopted as a performance measurement and management system. Its use has been extended beyond financial evaluation and supplemented with other measures such as customer satisfaction, internal processes, learning, and growth (Haes and Van Grembergen, 2005)

Balance Score Card is identified as an approach for the practical implementation of strategic alignment (Haes and Van Grembergen, 2004). The use of BSC as enabling and measurement tool for BIA is explored in extant literature (Ahuja, 2012, Umoh et al., 2012, Hu and C., 2006, Haes and Van Grembergen, 2004, Van Grembergen and Saull, 2001). In one of the studies (Van Grembergen and Saull, 2001), IT BSC is applied at the lower levels of an organisation referred to as development and operational BSC. These are then modelled to contribute to IT strategic BSC. IT Strategic BSC in turn contributes to business BSC. This is depicted in Figure 2.1

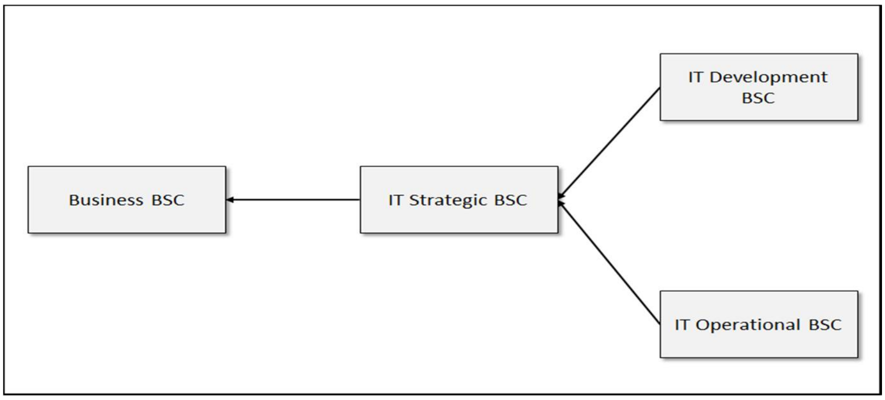

Figure 2.1: Cascade of Balanced Score Cards (Van Grembergen and Saull, 2001)

COBIT[1] and ITIL are ITSM frameworks that propose best practices that can contribute to achieving and sustaining BIA. Whereas COBIT defines what should be done and the different IT processes that have to be put in place, ITIL describes in detail how specific IT processes can be organised and managed for IT service management (Hardy, 2006, Haes and Van Grembergen, 2005). These best practices are important for improving return on investment (ROI), meeting regulatory requirements, risk mitigation, and optimization of cost (Hardy, 2006).

COBIT’s four domains, namely – plan and organize; acquire and implement; deliver and support; monitor and evaluate (Hardy, 2006) are applied in a number of ways to attain BIA. For example, a study investigating IT governance and BIA using COBIT's plan and organize domain indicate that organizations with the highest strategic maturity level also had the highest level of governance maturity in the domain studied (Hosseinbeig et al., 2011).

ITIL is considered the de facto standard for IT service providers (Marrone and Kolbe, 2011). A number of studies investigate the contributions ITIL can make to

[1] COBIT is a framework for governance and management of enterprise IT: www.isaca.org/cobit/

BIA (Kashanchi and Toland, 2006, Marrone and Kolbe, 2011, Esmaili et al., 2010). The effect of applying ITIL for the purpose of improving BIA can be observed at different levels of an organization. As such it is suggested that organizations implementing ITIL should pay attention to the operational and strategic level impact of the implementation. This is premised on the argument that as an organization's ITIL maturity increases, its BIA maturity advances (Marrone and Kolbe, 2011). In an exploratory study (Kashanchi and Toland, 2006), the benefits of ITIL are listed, namely, breaking down the communication barrier between business and IT, ability to support business strategy, improve IT strategy, improves availability and quality of services, and facilitates operational process consistency.

2.4.3 Alignment via Communication

The communication approach to business-IT alignment takes the 'social dimension' into consideration (Chen, 2010). It aims to narrow knowledge, cultural, and language gaps between IT and business stakeholders (Ralha and Gostinski, 2008, Strnadl, 2005, Mallick and Krishna, 1999). It is argued that a "common language" that helps IT stakeholders understand business and business stakeholders understand IT can lead to improved alignment maturity (van Der Zee and de Jong, 1999). This also ensures that there is mutual understanding of the business domain and learning both at organizational and personal levels. Domain ontology (Guarino, 1998) can be very useful in this regard by offering a medium for making business domain knowledge explicit and facilitating knowledge sharing.

Social dimensions such as shared understanding, common language, shared domain knowledge, organizational learning, and communication between business and IT stakeholders are identified as antecedents to alignment maturity (Schlosser, 2012, Reich and Benbassat, 2000, Preston and Karachanna, 2006, Campbell et al., 2005). These factors can influence both short-term and long-term alignment and are discussed briefly in this section. These factors are also discussed in the section on antecedents to BIA later in this chapter.

2.4.3.1 Common Language

Common language between business and IT is essential to ensure that business needs are expressed in a language that stakeholders mutually understand. When this is not the case the likelihood of misalignment between business and IT increases. This risk is heightened in domains with specialized terms and concepts that IT stakeholders may not have the requisite experience and training to grasp. Common language between business and IT is often cited as an antecedent to BIA (Luftman, 2000, Cumps et al., 2006a). It is also cited as a pre-requisite for building shared knowledge about IT and business domain, attaining organizational learning or achieving architectural competence (Luftman, 2000, Chen, 2008).

2.4.3.2 Shared Domain Understanding

Shared domain understanding is essential for quality interaction between business and IT. When there is shared domain understanding, it implies that there is sufficient understanding of concepts, processes, and procedures in the business domain. When this mutual understanding is lacking, business-IT alignment maturity can be severely hindered (Luftman et al., 1999, Coughlan et al., 2005). Shared domain knowledge is an antecedent to long-term alignment. It informs the suggestion that significant effort should be directed toward understanding shared domain knowledge (Reich and Benbassat, 2000).

2.4.3.3 Organizational and Personal Learning

One way to foster a common language between business and IT stakeholders and functions is through learning at personal and organizational levels. Organizational learning (inter/intra) is essential to improve communication and by extension BIA (Luftman and Rajkumark, 2007, Luftman, 2003). At the organizational level, learning requires shared interpretation, common language, and mutual understanding (Balhareth et al., 2012). On a personal level of learning, personnel in domains with specialized concepts and vocabularies can gain the knowledge required to communicate effectively thereby improving alignment maturity.

2.5 Levels of Business-IT Alignment

Business-IT Alignment can be observed at the three levels of an organization namely strategic, tactical, and operational. The distinction of alignment by organizational levels is informed by the difference in functions and activities at these levels. Strategic business-IT alignment refers to enterprise (organization/macro-alignment) wide alignment whereas operational and tactical refers to alignment at more granular (foundational/micro-alignment) levels of the organization. It can be argued that to actualize desired alignment at strategic level, alignment at operational and tactical levels have to be attained. Achieving business-IT alignment at the operational and tactical levels is not trivial. There is an increasing realisation that the path to sustainable BIA is by focusing on micro-alignment (Karpovsky and Galliers, 2015, Peppard et al., 2014, Whittington, 2014). Furthermore, the lack of practice-based guidelines for accomplishing operational and tactical alignment adds to this challenge (Tarafdar and Qrunfleh, 2009, Guiterrez et al., 2006).

2.5.1 Strategic Business-IT Alignment

Strategic alignment focuses on an organization’s future needs. It refers to the top-down, pull, vertical, executive management perspective of business-IT alignment (Simonsen, 1999, Solaimani and De Vries, 2010). The key elements of strategic alignment are business and IT strategy and plans. Strategic alignment is therefore concerned with synchronization between business and IT strategies and plans. Achieving strategic alignment requires the fine-tuning between business and IT plans, relationships and liaison between senior management from both sides of the organization (Johnson and Lederer, 2010, Tarafdar and Qrunfleh, 2009, Bon and Hoving, 2007, Benbya and McKelvey, 2006).

There is a near consensus of opinion in extant literature on the importance of strategic business-IT alignment. Despite the vast body of literature making the case for strategic business-IT alignment there is little in terms of guidance on how to achieve alignment in practice (Guiterrez et al., 2006). Most of extant literature focuses on strategic BIA (macro-alignment). An alternative perspective is to explore alignment from lower levels of the organization where execution of the plans and strategies are implemented (Coltman et al., 2015, Karpovsky and Galliers, 2015).

2.5.2 Operational Business-IT Alignment

Operational and tactical business-IT alignment are referred to as the horizontal level, push, bottom-up, and individual alignment (Benbya and McKelvey, 2006). The distinction between operational and tactical alignment is not as clear as the difference between strategic and functional (operational and tactical) alignment. The operational level is responsible for realization of goals within the chosen directions and organizational context (Bon and Hoving, 2007). Operational alignment aims to ensure effective and efficient IT support for an organization's day-to-day business processes and functions. Operational alignment is described as alignment between a particular information system and the related business domain (Hugoson and Pessi, 2011). Operational alignment focuses on the realization of an organization's strategies and plans (Reich and Benbassat, 2000, Tarafdar and Qrunfleh, 2009).

Operational business-IT alignment is achieved by business and IT operational and middle level management collaborating in planning, prioritization, and implementing actual projects of sizes ranging from small to mid-sized. It can be argued that processes for tactical alignment should facilitate operational level linkages between IT and business functions concurrently with application project implementation, technology choice, resource allocation, managerial goal synchronization, skills requirements, IT delivery capabilities, and governance of IT strategies (Tarafdar and Qrunfleh, 2009). This cross-functional interaction between business and IT fosters trust, mutual understanding, and domain knowledge sharing which are the key enablers to business-IT alignment.

2.5.3 Tactical Business-IT Alignment

From an organizational point of view the tactical level is responsible for designing and controlling the organization towards the realization of goals. The concept of tactical Business-IT Alignment was first highlighted in (Qrunfleh and Tarafdar, 2008). They argue that tactical BIA is necessary for ensuring that IT applications projects are implemented on time and deliver the planned and desired business benefits. Tactical BIA also facilitates an organization's allocation of its IT resources effectively. Five aspects of tactical BIA are highlighted:

1. *Communication*. This essentially involves communicating corporate strategies level to middle and lower level management. This is essential to keep IT stakeholders informed of planned applications, translation of corporate level strategy to IT infrastructure that actualize and facilitate these strategies. The information and knowledge from the communication also aids planning, selection of projects, and evaluation of requirements from the business.
2. *Balance*. This involves finding a balance between enterprise-wide IT standardization and process-specific customization sometimes required to gain competitive advantage or to address specific needs within the enterprise. This balance has an influence on IT acquisition and project implementation.
3. *Interaction*. There is expectedly more frequent interaction between middle and junior-level management from IT and business functions. This is essential for eliciting business needs, during implementing, and deploying of solutions to meet business needs. To achieve this, formal communications such as feedback about system requirements, systems deficiencies, user problems, and informal communication is essential.
4. *Governance*. In setting up structures and positions for managing initiatives and mapping requirements to solutions, 'liaison positions' are advocated. Also, an 'executive sponsor', responsible for championing business initiatives and linking these to specific operational-level business benefits and obtaining management buy-in is advocated.

5. *Project Management*. Project planning and implementation is the stage at which organization strategies from business and IT are executed. This requires frequent communication and interaction between stakeholders from business and IT functions. BIA at this level involves matching IT project deliverables business and IT strategies. This is actualized in part through agility and flexibility in responding to changing requirements and resource issues.

The outcomes of tactical BIA are identified as being the enabling execution of business strategy and better relationships between IT and business managers at the operational level and execution of IT strategy (Qrunfleh and Tarafdar, 2008). A summary of the three levels at which BIA can be is presented in Table 2.5.

2.6 Antecedents to Business-IT Alignment

Considerable research focus is directed on pre-requisites, factors, and conditions necessary for achieving and sustaining BIA. Qualitative and quantitative approaches have been used in these studies to validate how strongly these antecedents influence business-IT alignment maturity. The instrument used for data collection in these studies are mainly questionnaires and surveys with respondents answering questions aimed at gauging their opinion on pre-identified items embodied in models/constructs developed for the studies. The responses are analysed and validated using statistical analysis to gauge alignment maturity.

The antecedents that have been identified have varying degrees of support in extant literature. Antecedents like shared understanding and domain knowledge seems to be widely accepted (Chan et al., 2006, Chan and Reich, 2007). Other antecedents such as structures and processes, IT and business management sophistication does not have unanimous support (Chan et al., 2006, Almajali and Dahalin, 2011). In Table 2.6, antecedents identified in extant literature are presented.

Table 2.5: Levels of Business IT Alignment

	BIA at Strategic Level	BIA at Tactical & Operational Levels
Perspective	Top-Down	Bottom-Up
Overview/Scope	Strategic alignment helps organizations meet future IT needs (Chen, 2010)	Tactical alignment allows the organization to allocate its IT resources effectively. Operational alignment ensures effectiveness and efficiency of IT in supporting the organization's operations on a daily basis (Chen, 2010)
Approach Aims and Objectives	Strategic alignment approaches aim to improve organisational performance, enhance efficiency and maintain or gain competitive advantage (Gutierrez and Serrano, 2008)	Tactical/Operational alignment aim to improve alignment through the day-to-day operations of application and infrastructure and interaction between business users and IT stakeholders. The focus is on realization of the organization's strategies and plans (Tarafdar and Qrunfleh, 2009, Gutierrez and Serrano, 2008)
Antecedents	Examples include shared domain knowledge, communication between business and IT executives, connections between business and IT planning, and history of Successful IT delivery	Resource allocation imperatives, project selection and execution priorities, and technology-choice decisions of the IT function are alignment with those of the other functions at the operational level (Qrunfleh and Tarafdar, 2008)
Processes	1. Synchronizing business planning and IT planning such that strategic IT plans support business plans. 2. Exploiting IT-based strategic opportunities. 3. Proactive influence of the CIO in strategic planning (Tarafdar and Qrunfleh, 2009)	1. Alignment at the level of projects. 2. Aligning decision-making processes of the IT function and other departments. 3. Balancing firm-wide technology standardization with process specific customization. 4. Formal and informal IT-business communication. 5. Alignment at the level of IT skills (Tarafdar and Qrunfleh, 2009)
Outcomes	1. Achievement of organizational objectives. 2. Improved managerial outcomes driven by regular communication and common shared goals. 3. Organizational performance improvement (Chan et al., 2006)	1. Enable execution of business strategy. 2. Enable better relationships between IT and business managers at the operational level. 3. Enable execution of IT strategy (Qrunfleh and Tarafdar, 2008)

Table 2.6: Antecedents to Business IT Alignment

Antecedents	References
Enablers: 1.Senior executive support for IT. 2. IT involved in strategy development. 3. IT understands the business. 4. Business-IT partnership. 5. Well-prioritized IT projects. 6. IT demonstrates leadership. Inhibitors: 1. IT/business lack close relationships. 2. IT does not prioritize well. 3. IT fails to meet commitments. 4. IT does not understand business. 5. Senior executives do not support IT. 6. IT management lacks leadership	(Luftman et al., 1999)
1. Top management is committed to the strategic use of IT. 2. IS management is knowledgeable about business. 3. Top management has confidence in the IS department (Related to 1). 4. The IS department provides efficient and reliable services to user departments.5. There is frequent communication between users and IS department. 6. The IS staffs are able to keep up with advances in IT.7. Business and IS management work together in partnership in prioritizing applications development. 8. Business goals and objectives are made known to IS management. 9. The IS department is responsive to user needs (Related to 4). 10. Top management is knowledgeable about IT. 11. The IS department often comes up with creative ideas on how to use IT strategically (Related to 6). 12. The corporate business plan is made available to IS management (Related to 8)	(Teo and Ang, 1999)
1. Shared domain knowledge. 2. Communication between business and IT executives. 3. Connections between business and IT planning. 4. Successful IT history	(Reich and Benbassat, 2000)
1. CEO and CIO have a strong working relationship. 2. Business and IS plans are closely linked. 3. IS personnel participate in business planning. 4. IS projects have business sponsors. 5. IS personnel make lateral short- or long-term transfers into business partner areas 6. Incentive/compensation bonus schemes exist	(Chan, 2002)
1. Information intensity of the value chain. 2. The CIO participates in business planning. 3. The CEO participates in IT planning. 4. The IT plan reflects the business plan	(Kearns and Lederer, 2003)
1. Shared domain knowledge. 2. Prior IS success. 3. Organizational size	(Chan et al., 2006)
1. Integrating IT planning with business planning. 2. Maintaining effective communication channels. 3. Developing strong relationships between IT and business. 4. Institutionalizing the culture of alignment	(Huang and Hu, 2007)
1. IT Unit Structure. 2. Shared Domain Knowledge. 3. Successful IT History. 4. Relationship Management	(Yayla and Hu, 2009)

2.7 Measuring Business-IT Alignment

Measuring BIA maturity is an important aspect of BIA research and the subject of numerous extant literature (Haes et al., 2010, Sledgianowski et al., 2006). Objective measurement of BIA maturity is difficult due to the dynamics and complexities of BIA. Most of extant literature follow a qualitative approach to BIA measurement (Luftman et al., 2008) using questionnaires (Kearns and Lederer, 2003, Bergeron et al., 2004), a few use quantitative metric-driven methods (Simonin et al., 2007, Gmati et al., 2010), and others follow frameworks (Luftman, 2000, Luftman and Brier, 1999). One of the first models developed for measuring BIA is the Strategic Alignment Maturity Model (SAMM) (Luftman, 2000), based on the capability maturity model (CMM) for alignment. The alignment levels and pathways are depicted in Figure 2.2.

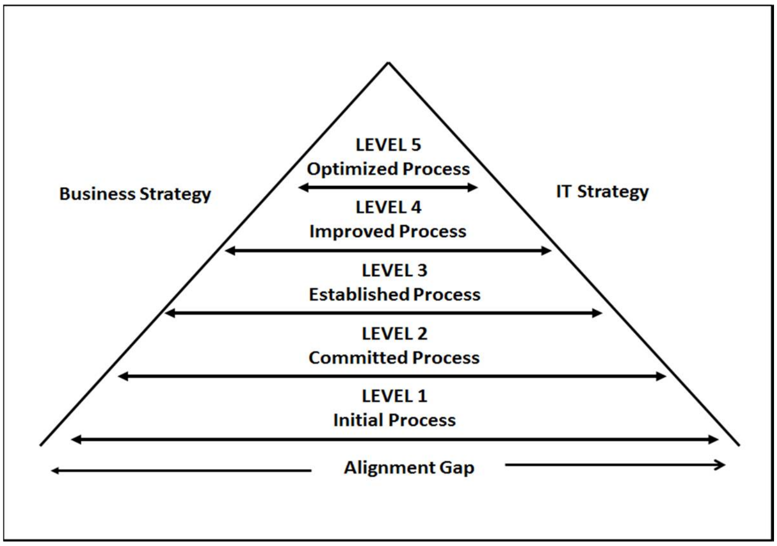

Figure 2.2: Climbing the Strategic Alignment Maturity Model

The model identifies levels of maturity that an organisation's alignment can be placed and pathways and actions required to attain an optimized level of alignment. Other models have since been developed, also based on the strategic alignment model (SAM) such as the generic framework (Maes, 1999). This model has been combined with the Integrated Architecture Framework (IAF) (Goedvolk, 1999) resulting in the unified framework (Maes et al., 2000).

The generic maturity model is adopted for gauging BIA maturity in the SAMM. This generic model is also used for ranking IT governance maturity. The model is presented in Table 2.7. SAMM has six maturity variables for gauging BIA in organizations namely. communication, competency/value measurement, governance, partnership, scope and architecture, and skills (Luftman, 2000). SAMM is depicted in Figure 2.3.

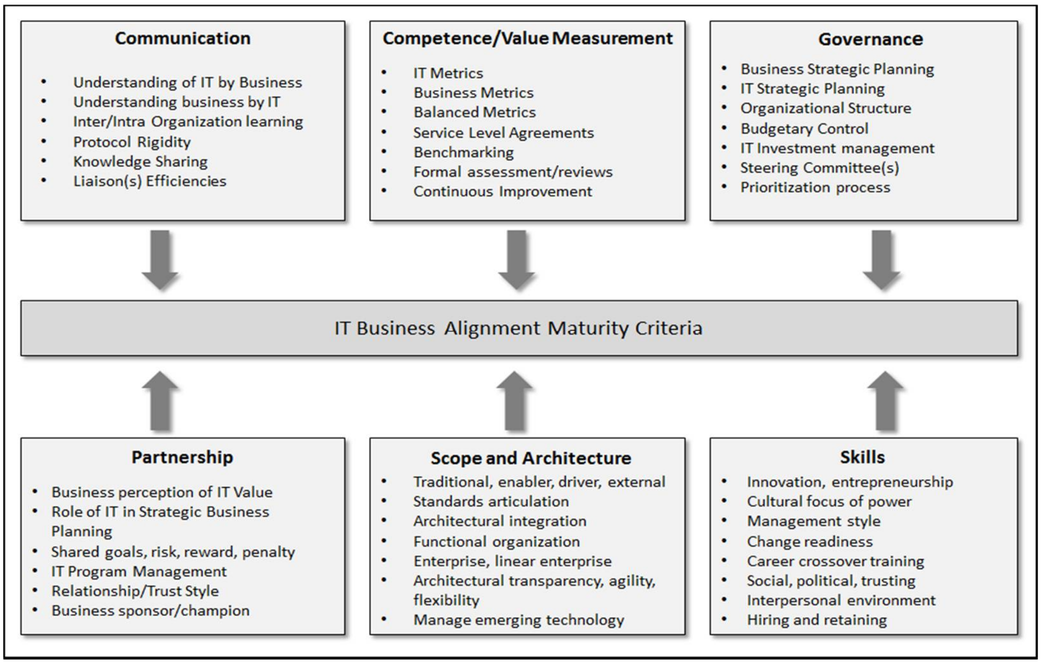

Figure 2.3: Strategic Alignment Maturity Model

SAMM is conceptualized as a strategic level framework. Its measures are therefore better suited to the strategic level and less suitable for operational and tactical level without some modifications. Furthermore, literature on operationalizing SAMM in practice is lacking. This leaves a gap that that can be filled by making adaptations to it or by combining it with other frameworks for functional level alignment.

Table 2.7: Generic Maturity Model (COBIT)

Maturity level	Description
0. Non-existent	Complete lack of any recognizable processes. The enterprise has not even recognized that there is an issue to be addressed
1. Initial/ad hoc	There is evidence that the enterprise has recognized that the issues exist and need to be

	addressed. There are, however, no standardized processes; instead there are ad hoc approaches that tend to be applied on an individual or case-by-case basis. The overall approach to management is disorganized
2. Repeatable but intuitive	Processes have developed to the stage where similar procedures are followed by different people undertaking the same task. There is no formal training or communication of standard procedures, and responsibility is left to the individual. There is a high degree of reliance on the knowledge of individuals, and therefore errors are likely
3. Defined process	Procedures have been standardized and documented, and communicated through training. It is mandated that these processes should be followed; however, it is unlikely that deviations will be detected. The procedures themselves are not sophisticated, but are the formalization of existing practices
4. Managed and measurable	Management monitors and measures compliance with procedures and takes action where processes appear not to be working effectively. Processes are under constant improvement and provide good practice. Automation and tools are used in a limited or fragmented way
5. Optimized	Processes have been refined to a level of good practice, based on the results of continuous improvement and maturity modelling with other enterprises. IT is used in an integrated way to automate the workflow, providing tools to improve quality and effectiveness, making the enterprise quick to adapt

2.8 Research Trends in Business-IT Alignment

The focus of BIA research over a period from the 1970s to 2000s is depicted in a time graph (Silvius, 2007). It shows that BIA research originated from traditional IT planning, transitioned to modern IT planning, then to BIA. An extension of this graph to highlight the focus and evolution of BIA research in recent times is presented in Figure 2.4.

In the 2000s research focus was on extending the Strategic Alignment Model (Henderson and Venkatraman, 1993). There was also focus on developing maturity models to gauge BIA maturity. This is evidenced in extant literature on models that are adaptations of or based on SAMM (Luftman, 2000). There is literature focusing on identifying and verifying antecedents to BIA. Exploring BIA at different levels of an organization also attracted research interest. Post the year 2000 the focus has been on investigating how other concepts and frameworks such as ITIL influence or can be applied in conjunction with BIA and recently on micro-alignment.

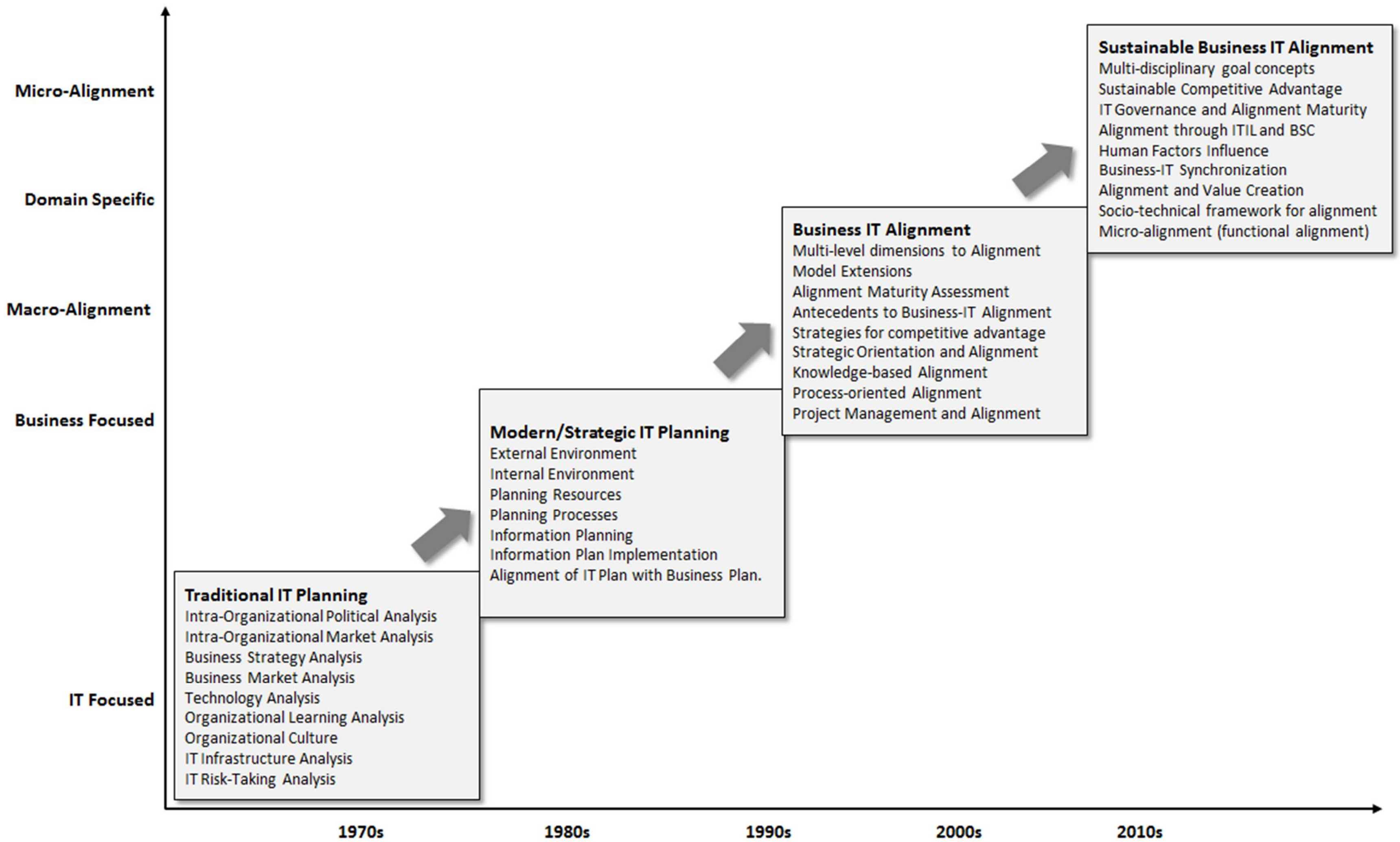

Figure 2.4: Evolution of BIA Research Focus

2.8.1 Theories applied to Business-IT Alignment Research

Research on BIA has been influenced by theories in the field of strategy. Much of the research materials on strategy were developed in the 1960s. This includes resource-based view (RBV), theory of dynamic capabilities, co-evolutionary and complexity theory, and lately, micro-foundational theory.

RBV posits that the firm’s internal resources are the primary predictors of superior financial performance (Wernerfelt, 1984). As argued in the chapter one, BIA is a means to gain completive advantage (Kearns and Lederer, 2003). By applying the RBV theory, competitive advantage can be obtained from strategic IT alignment when it represents a complex organizational process that is both heterogeneous and immobile (Kearns and Lederer, 2003, Peteraf, 1993, Wernerfelt, 1984, Barney et al., 2001).

The theory of dynamic capability posits that the development of dynamic capabilities is limited by a firm's existing resources and is shaped by its current market position and history of developing past resources (Grant, 1996, Teece et al., 1997, Montealegre, 2002). The theory, which is based on the resource-based view, builds on the strength and competency of resource reconfiguration in organizations. The theory has been applied to business strategy (Eisenhardt and Martin, 2000, Zollo and Winter, 2002, Winter, 2003).

The argument for the application of the theory of dynamic capabilities is that it is critical for the creation and strength of IT resources and therefore can positively influence the alignment process and its future implementation success. The theory of dynamic capabilities approach is applied in a case study surmise that dynamic capabilities, obtained though learning, integration and transformation processes, positively influence the alignment process and the achievement of ‘implemented alignment’ (Chen et al., 2008).

In extant literature, business-IT alignment is often conceptualized as a state or as an outcome. This assumes that if the top-down or bottom-up perspectives are followed an organization can attain a state of alignment. This perception overlooks the continuously changing and evolving nature of organizations and the environment in which they function. The co-evolutionary and emergent view of alignment takes this dynamic nature of alignment into consideration. One of the reasons alignment has been elusive can be attributed to this (Benbya and McKelvey, 2006, Hugoson and Pessi, 2011, Peppard and Brue, 2003).

Co-evolutionary theory can be applied to alignment at the three organizational levels (strategic, operational and tactical/individual). At the strategic level it involves co-evolution of business and IT strategy. At the operational level, co-evolution of business and the IT units/departments is required. To achieve this common language, shared understanding and collaborative partnership between business and IT must be established. Co-evolutionary alignment at the tactical/individual level requires closer interaction between business and IT in the systems development process. The requirements and needs from individuals within both functions are aligned. The shared understanding and common language at the operational level is also desired at this level (Benbya and McKelvey, 2006).

Co-evolutionary theory is applied to BIA as a promising alternative to other theories in acknowledgment of the complex nature of BIA. The unique feature of this theory is its causal loop that is a feedback mechanism. This is significant as it caters for the evolving and constantly changing environment in which new capabilities must evolve in tandem (Peppard and Brue, 2003).

There are other theories that are applicable to BIA. These include punctuated equilibrium model (Sabherwal et al., 2001) and micro-foundations theory of management (Teece, 2007, Devinney, 2013). Punctuated equilibrium model considers dynamic alignment and the pace of IT change as business strategy evolves. Micro-foundation theory from the perspective of alignment considers how functional alignment aggregates to strategic level alignment (Coltman et al., 2015).

2.9 Relationship between BIA and RE

The relationship between RE and BIA in various context is explored in existing research (Bleistein et al., 2005, Bleistein et al., 2004, Ullah and Lai, 2013, Ullah and Lai, 2011, Chebrolu and Ness, 2013, Salgado et al., 2013). The requirements engineering process in software development projects within an organization should be in alignment with the organization’s business strategy. The importance of the requirements analysis capturing strategic business objectives, activities and processes by which those objectives are to be achieved has been emphasized. It can be argued that by getting requirements right, organizations can also get alignment right (Babar et al., 2007). While extant literature acknowledges the link between requirements engineering and BIA they fall short by not defining or proposing the means by which this relationship can be leveraged practically to attain sustainable BIA throughout the SDLC.

The knowledge and language gaps between business and IT stakeholders directly or indirectly affects communication and interaction between business and IT stakeholders adversely. The consequence of this gap is insufficient requirements gathering and misunderstanding requirements. This makes a strong case for a framework that can bridge knowledge and language gaps, guide the requirements elicitation process, and facilitate effective management of the elicited requirements throughout the SDLC and the stages of the entire project. This study contributes to addressing these gaps by proposing a structured approach to elicitation of high quality requirements, management of changes to these requirements, reuse of historic requirements, constraints, and existing IT artefacts, and closing knowledge and language gaps identified as inhibitors of BIA.

2.10 Requirements Engineering Activities

Requirements engineering activities namely elicitation, analysis, specification, validation, and verification present an opportunity to drive sustainable BIA. These RE activities form an iterative process, which is repeated as requirements change and new requirements are added especially in agile and iterative SDLC models. Changes and additions to requirements have to be rigorously managed to ensure the quality of the RE process is maintained. Modifications to requirements must be done in a transparent manner. The effects and implications of changes to validated requirements and addition of new requirements must be clearly communicated to and understood by all stakeholders. Changes and additions can occur at any point within the requirements engineering process. The activities that support these changes and additions to requirements are collectively referred to as *Requirements Management* and can also influence BIA and are explored later in this chapter.

The flow between the four requirements engineering activities is depicted in Figure 2.5. These activities are discussed in this section.

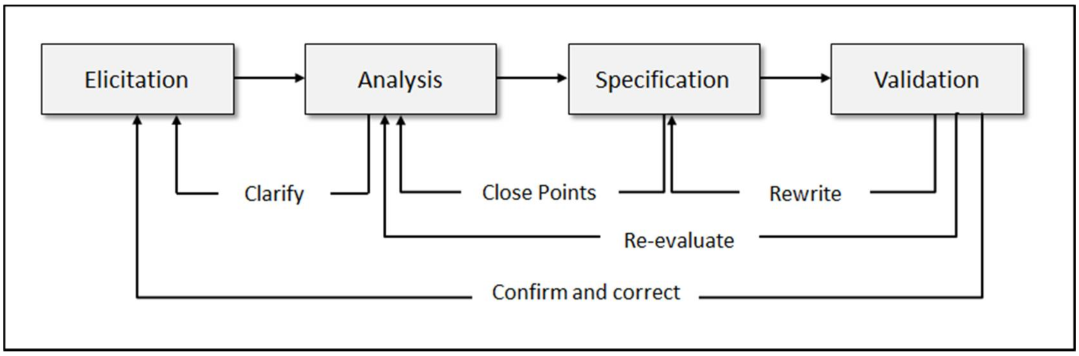

Figure 2.5: Requirements Engineering Processes

Requirements elicitation is a collaborative, analytical, and communication-intensive process especially when tacit requirements are to be elicited (Gacitua et al., 2009). As such, for requirements elicitation to be effective, knowledge of the business domain, business problem, and of the organization are pre-requisites. The elicitation process should not merely be an exercise to gather requirements and document business user comments, rather, it should be a rigorous process to collect, discover, extract, and refine requirements. The output of the elicitation process is the discovery of business, user, functional, and non-functional requirements. Requirements elicitation is challenging and error-prone and therefore requires considerable thought and effort (Sommerville and Sawyer, 1997a, Vliet, 2007, Wiegers and Beatty, 2013).

There are a number of requirements elicitation methods that can be adopted in software development projects. These include prototyping, structured surveys, ethnography, interviewing, brainstorming, and job shadowing. Each of these methods has pros, cons, and context for which they are best suited. The elicitation method adopted for a particular project can have an impact on the interaction and collaboration between business and IT stakeholders during the project.

The requirements analysis activity is aimed at organizing, prioritizing translating business needs into technical requirements for software elements, and resolving conflicts, overlaps, omissions, and inconsistencies in the elicited requirements. These activities require collaboration and negotiation between stakeholders from business and IT functions and presents an opportunity to drive sustainable BIA. Requirements analysis is expensive in terms of resources and time to read documents and analyse the implications of the statements in the documents. This makes a strong case for knowledge-based tools and frameworks that can leverage domain knowledge to guide the requirements analysis process (Sommerville and Sawyer, 1997a, Aurum and Wohlin, 2005, Fairley, 2009, Tsui et al., 2014).

The objective of the requirements specification activity is to capture functionalities of the intended software, interfaces with other system elements, and constraints on its operation in formal language and notation. Decisions on accepting or rejecting requirements are based on the requirements specification and constraints. The specification activity ends with producing a software requirements specification (SRS). SRS documents should be precise, easy to understand, consistent, and complete with no ambiguities, contradictions or conflicts (Pressman, 2005, Endres and Rombach, 2003, Agarwal et al., 2010, Sommerville, 2011).

The requirements validation activity involves checking the defined set of requirements for possible problems. It requires the collaboration of both business and IT stakeholders. Requirements validation is a critical activity in requirements engineering. Errors in SRSs can result in extensive rework costs when discovered during the development phase or after deployment in production. To avoid reworking and retesting the software and the huge associated costs, it is essential that requirements validation checks be rigorously carried out. This includes checking for additional or different functions required to fulfil a requirement (validity); checking for contradictory constraints or different descriptions for the same functionality (consistency); checking to ensure all functions and constraints are accurately captured (completeness); checking to verify feasibility of requirements (realism); checking to ensure adequate tests are captured to verify fulfilment of the requirements (verifiability) (Sommerville, 2011, Sommerville and Sawyer, 1997a).

2.10.1 Elicitation Methods in Requirements Engineering

The requirements elicitation activity is a knowledge and communication intensive process and is rarely problem-free. A number of problems can occur during requirements elicitation have been identified (Lee, 2013). These problems can be attributed to two major causes. Firstly, there are language and knowledge gaps between business and IT stakeholders, IT personnel having poor knowledge of business problem domain, business and IT stakeholders speaking different languages, and business stakeholders having poor understanding of technical capabilities and limitations. Secondly, there are issues related to the volatile and evolving nature of requirements especially in rapidly evolving and highly regulated industries such as financial services.

This relates to the language and knowledge gap between business and IT stakeholders leading to misalignment, highlighted in this chapter. In this section methods commonly used for requirement elicitation are discussed with emphasis on how they influence business and IT alignment factors. There are other elicitation methods not covered in this section such as introspection, questionnaires, system interface analysis, user interface analysis, and focus groups that can be used. A comparison of the elicitation methods discussed with respect to factors that influence business-IT alignment is presented in Table 2.8.

Table 2.8: Analysis of Requirements Elicitation Methods

Factor	Detail	Brainstorming	Document Analysis	Ethnography	Interviews	Observation	Proto-typing	Scenario	Workshops
Requirements Ambiguity	Unclear business need	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable
Requirements Ambiguity	Discovery of overlooked constraints	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable
Deployment Strategy	Big-bang delivery	Suitable	Maybe	Unsuitable	Suitable	Suitable	Suitable	Maybe	Suitable
Deployment Strategy	Phased delivery	Suitable	Maybe	Unsuitable	Suitable	Suitable	Suitable	Suitable	Maybe
Domain Knowledge	Gain understanding of domain	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Unsuitable	Suitable
Domain Knowledge	Domain knowledge required	Unsuitable	Unsuitable	Unsuitable	Maybe	Suitable	Unsuitable	Maybe	Unsuitable
Domain Knowledge	Technologically-aware stakeholders	Suitable	Suitable	Unsuitable	Suitable	Suitable	Suitable	Suitable	Maybe
Project Cost	Large budget	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Maybe	Suitable
Project Cost	Small budget	Suitable	Maybe	Unsuitable	Suitable	Suitable	Maybe	Suitable	Maybe
SDLC Model	Agile & Iterative Methodologies	Suitable	Unsuitable	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable
SDLC Model	Waterfall Methodology	Maybe	Maybe	Suitable	Suitable	Maybe	Maybe	Maybe	Suitable
Project Scope	Large-scale (enterprise) project	Suitable	Suitable	Suitable	Suitable	Suitable	Maybe	Maybe	Suitable
Project Scope	Medium-scale project	Maybe	Maybe	Maybe	Suitable	Suitable	Suitable	Suitable	Suitable
Project Scope	Low-scale project	Maybe	Maybe	Unsuitable	Maybe	Suitable	Suitable	Suitable	Unsuitable
Number of Stakeholders	Large number of stakeholders	Suitable	Suitable	Suitable	Suitable	Maybe	Suitable	Suitable	Suitable
Number of Stakeholders	Small number of stakeholders	Suitable	Maybe	Suitable	Suitable	Suitable	Suitable	Suitable	Unsuitable
Project Purpose (Type)	Enhancement of existing system	Suitable	Suitable	Suitable	Suitable	Suitable	Unsuitable	Maybe	Suitable
Project Purpose (Type)	New system	Unsuitable	Suitable	Unsuitable	Suitable	Suitable	Suitable	Suitable	Suitable
Project Purpose (Type)	Replacement of existing system	Suitable	Suitable	Unsuitable	Suitable	Maybe	Unsuitable	Suitable	Suitable
Stakeholder Location	Mostly co-located	Suitable	Suitable	Unsuitable	Suitable	Suitable	Suitable	Suitable	Suitable
Stakeholder Location	Geographically dispersed	Maybe	Suitable	Unsuitable	Maybe	Unsuitable	Suitable	Maybe	Maybe
Time to Market	Short time-to-market	Suitable	Unsuitable	Unsuitable	Suitable	Suitable	Suitable	Maybe	Maybe
Time to Market	Long time-to-market	Suitable	Suitable	Suitable	Suitable	Suitable	Suitable	Maybe	Suitable
Requirements Reuse	Reuse historic requirements	Unsuitable	Maybe	Maybe	Maybe	Maybe	Maybe	Suitable	Maybe

Requirements elicitation by brainstorming involves stakeholders meeting in sessions to discuss in informal and unstructured settings at a high level to produce mission statements and initial requirements for the proposed software. Brainstorming is an elicitation method well suited for agile SDLC models (Zowghi and Coulin, 2005, Chemuturi, 2013). It facilitates close interaction between business and IT stakeholders during requirements elicitation.

Also referred to as domain analysis, document analysis when used for requirements elicitation for legacy systems can help identify functionality that can be retained and those that are obsolete. It is also useful for capturing domain knowledge and identifying reusable artefacts. Using this method, industry standards, regulatory demands, and aspects of requirements that are overlooked by business can be discovered by stakeholders. This method is typically used with other elicitation methods (Zowghi and Coulin, 2005, Wiegers and Beatty, 2013).

Ethnographic requirements elicitation methods are quite effective when new systems are needed to address existing problems with processes and procedures. They are also useful for identification of social patterns and complex relationships between human stakeholders. In contexts where the interaction between different users of the system is critical, ethnographic elicitation methods are preferred (Sommerville et al., 1993, Chemuturi, 2013).

Interviewing is the most commonly used requirements elicitation technique. Interviews are particularly useful in SDLC models that emphasize user involvement such as agile and RAD (Pressman, 2005, Sommerville, 2011, Wiegers and Beatty, 2013, Chemuturi, 2013). Interviewing business stakeholders can provide an opportunity for technology stakeholders to gain a better understanding of the business domain and problem space.

The observation elicitation method involves IT personnel observing business users performing the functions for which the desired software is to support. Observation can either be silent or interactive. In certain business scenarios, users cannot be interrupted while they perform the functions for which the software is required. In such cases, silent observations are more appropriate. Interactive observations allow the IT stakeholder to interrupt the user while performing the function to ask questions (Wiegers and Beatty, 2013, Chemuturi, 2013).

Prototyping involves producing an initial version of the software that can be used to demonstrate concepts to stakeholders. It is especially useful when requirements are incomplete, ambiguous or the business problem is not fully understood at the inception of the development lifecycle. Prototyping by its nature is best suited to agile, RAD and iterative SDLC models (Pressman, 2005, Zowghi and Coulin, 2005, Sommerville, 2011, Wiegers and Beatty, 2013, Chemuturi, 2013).

Scenarios are a commonly used requirement elicitation method. The advantage of using scenarios is that it facilitates provision of details in addition to abstract models of the proposed software. Scenarios are useful in the requirements engineering process beyond requirements elicitation. For example, scenarios can help in testing models and specifications during requirements validation (Sutcliffe, 2003, Sutcliffe et al., 1998, Zowghi and Coulin, 2005).

Workshops offer an excellent platform for rapid elicitation and capture of requirements. Workshops facilitate transfer of domain knowledge from business stakeholders to IT stakeholders. Workshops are typically used with other requirements elicitation methods such as scenarios. In this context, workshop participants can be separated into groups to develop scenarios. The groups can then be combined to go through the scenarios and generate requirements for the software. By its nature, workshops are an excellent elicitation method for iterative and agile SDLC models (Chemuturi, 2013, Wiegers and Beatty, 2013, Zowghi and Coulin, 2005).

2.10.2 BIA and Requirements Engineering Activities

The RE activities discussed earlier, namely elicitation, analysis, specification and validation, can be leveraged to attain sustainable business-IT alignment. Organizational processes and procedures for these activities can be established or geared towards this aim. In this section, the link between these individual RE activities and business-IT alignment are explored further.

Factors that drive the choice of elicitation method such as project size, project purpose, resources available for the project, and knowledge and experience of the stakeholders, especially in knowledge-intensive business domains have been highlighted. These factors and the eventual choice of elicitation methods for projects can have an impact on business-IT alignment maturity. It is therefore important that in choosing elicitation method(s), the interaction and collaboration between business and IT stakeholders is taken into consideration. Requirements elicitation methods have been comparatively analysed.

Analysis of requirements post-elicitation should be carried out with the aim of facilitating sustainable business-IT alignment. Requirements should be organized and prioritized by business priority whilst taking technical constraints and limitations into consideration. Translation of business needs into technical requirements should be in language that business and IT stakeholders can mutually understand. IT stakeholders should have some business domain knowledge and business stakeholders should possess a level of technical awareness. This can quicken the resolution of requirements conflicts, overlaps, omissions, and inconsistencies.

Post-analysis activities can also be performed with sustainable business-IT alignment in mind. The use of formal notations to specify requirements and the production of technical specifications require business domain knowledge and understanding of the notations used. Since decisions on acceptance or rejection of requirements are based on these specifications, it should be expressed in language and notations business stakeholders can readily grasp. This is critical to get their contributions and avoid misunderstanding that can affect the quality and validity of the requirements and IT artefacts developed based on the requirements.

Requirements validation and verification presents an opportunity to address possible problems with the requirements. The collaboration between business and IT stakeholders for this activity is critical to avoiding errors, which have severe tangible and intangible impacts on project outcomes. The validity, consistency, completeness, realism and verifiability tests that are recommended, when conducted properly, can facilitate business-IT alignment.

In summary, the involvement of business stakeholders in requirements engineering activities present an opportunity for establishing processes and procedures for achieving sustainable business-IT alignment. This argument is in line with the findings of the CHAOS group that user involvement in the SDLC and PMLC can contribute to more successful software development project outcomes (Schwalbe, 2014). This can also extend to better aligned business and IT functions.

2.11 Requirements Engineering Work Products

A work product models something of value that is produced, used, modified or destroyed during the performance of a process. The main work products of RE include business case, business rules, requirements, use cases, analyses such as customer analysis, user analysis, market analysis, and technology analysis. The work product explored further in this section is requirements. An ideal outcome of a requirements engineering endeavour is production of high quality requirements that meet current needs and can be reused as template for future needs. The focus of this section is requirements quality, patterns, and reuse, concepts central to business-IT alignment are reviewed in this chapter. This section ends with a consideration of the influence these concepts can have on business-IT alignment.

2.11.1 Requirements Quality

The link between requirements quality and the quality of the produced software artefact is one that has attracted wide research interest (Kaplan and Norton, 1992, Knauss and El Boustani, 2008, Sepehri et al., 2009, Carlson and Laplante, 2014).

Findings indicate that high quality requirements can lead to software development project success. In highly specialized domains such as financial services where efficiency and accuracy is critical, careful attention should be paid to requirements quality. This informs the research interest in the link between software quality and business-IT alignment (Haigh, 2010).

To achieve sustainable BIA, it is necessary to develop a means to consistently obtain high quality requirements for software projects. There should also be a framework for evaluating and measuring the elicited requirements to ensure that the quality is maintained throughout the requirements engineering and management process as new requirements are added and existing requirements modified.

The identification and validation of requirements and software product quality metrics has attracted research interest. Metrics and processes for requirements and software quality measurement tend to be subjective (Tian, 2005, Sommerville, 2011). The IEEE standard for requirement document and requirements quality, IEEE Std. 830-1998 (1998), provides guidelines that can be used for developing a requirements quality assurance framework. It can therefore be argued that ensuring BIA through high quality requirements depends on the appropriateness of the elicitation method, participation of relevant stakeholders, and effective requirements management practices. Requirements document qualities are summarized in Table 2.9.

Table 2.9: Requirement Document Quality

Quality Factor	Quality Criteria
Unambiguity and Consistency	Individual requirements consistent and unambiguous; Individual requirements do not conflict; Individual requirements uniquely identified
Clear Structure	Readability; Comprehensive and clearly structured
Modifiability and Extendibility	Easy to extend and modify (change, alter, add, remove as project progresses), version controlled
Completeness	Contain relevant requirements. All possible inputs, influential factors and required reactions described. Errors and exceptions described
Traceability	Documents such as business process model, test plans, or design plans created in current and previous development phases traceable

2.11.2 Requirements Reuse and Patterns

Analysis of recurring requirements in a given business domain can reveal certain patterns such as similarity in purpose, scope, and relationships with other requirements for a given business function. This implies that requirements can be reused as is or as the basis for new requirements (Robertson and Robertson, 2012). This informs the research interest in requirements patterns and reuse, their application in practice, and potentials for business-IT alignment maturity uplift.

One of the main benefits of using requirements management tools is facilitating requirements reuse. The repository feature of these tools allows for persisting requirements and facilitates reusing them in multiple projects or subprojects. To achieve this, requirements can be grouped into logical categories that fit descriptions of defined functions of the product (in this case software) then stored and referenced when necessary. This reduces requirements duplication, time-to-market, development cost, facilitates quicker elicitation, serves as a medium of knowledge management, and facilitates learning for both IT and business stakeholders (Goldin and Berry, 2013). This lends credence to the argument that reusability is critical to improving software development productivity and quality (Sommerville, 2011).

Historically, requirements reuse has been considered from the perspective of analogy (Lam et al., 1997, Finkelstein, 1988, Maiden and Sutcliffe, 1993b, Maiden and Sutcliffe, 1991, Massonet and Van Lamsweerde, 1997), case-based reasoning (Maiden and Sutcliffe, 1993a, Lopez De Mantaras et al., 2005) and generic modelling (Ryan and Mathews, 1993, Miriyala and Harandi, 1991). In more recent studies, it has been examined in the context of concepts like software product lines (Niu et al., 2014, Dehlinger and Lutz, 2008, Gomaa and Olimpiew, 2008), service-oriented (Lewis et al., 2005, Tsai et al., 2007) and component-based development (Gomes and Bento, 2000, Sharma et al., 2007).

Requirements reuse in product lines can be said to be analogous to a template for mass manufacturing of products. Each software product is minted from the same set of requirements and artefacts. Carnegie Mellon University's Software Engineering Institute (Northrop and Clements, 2012) provide a set of guidelines on what qualifies as a software product line is. They however posit that software product lines go beyond just reusing requirements.

Of the myriad of approaches to reuse, patterns stand out (Franch et al., 2010). The concept of requirements patterns can be said to mirror design patterns by the 'Gang of Four' (Gamma et al., 1994) that has been successfully applied in software engineering. Patterns are based on the observation that software engineering problems are not entirely unique and that there exist solutions appropriate for these recurring problems. Patterns therefore exist to provide best in class solutions to recurring software engineering problems.

Requirements patterns (Withall, 2007, Robertson and Robertson, 2012), provide templates for writing particular types of requirements for repeatable domain needs. They facilitate writing of higher quality requirements in shorter times and with less effort. Requirements patterns are conceptualized as best in class patterns to follow to define requirements for recurring business needs. The advantages of applying requirements patterns are elicitation of high quality requirements and better specification of requirements (Franch et al., 2010, Hoffmann et al., 2004).

2.11.3 Alignment and Requirements Work Products

Requirements have been identified as a work product of RE and it has been argued that there is a link between requirements engineering and business-IT alignment. It has also been argued that producing high quality requirements, leveraging requirements patterns in the business domain, and requirements reuse can potentially enable business-IT alignment. Conversely, poor quality requirements and not reusing requirements can potentially inhibit business-IT alignment. The impact requirements quality, patterns, and reuse can have on business-IT alignment is explored in this section.

Requirements quality is a critical factor in the link between requirements engineering and business-IT alignment. Poor requirements quality manifested in ambiguity, incompleteness, inaccuracy, high volatility, and weak version controls can lead to misunderstanding of the requirements, inaccurate estimation of tasks by developers. It is a leading cause of rework or discarding of IT artefacts and loss of time, resources, and competitive advantage.

Requirements patterns can be leveraged to improve business-IT through domain knowledge sharing, learning, and communication. Requirements patterns can directly or indirectly influencing business-IT alignment. These include grouping of recurring requirements by business function, improvement in requirements linkage, more effective communication about requirements changes, improved organizational and personal learning, better domain knowledge sharing, and making domain knowledge explicit.

Requirements reuse has potential benefits for business-IT alignment. The time and resource savings obtained through reuse can be channelled towards other development and testing tasks. Requirements reuse can also facilitate communication between business and IT stakeholders by providing a reference point used in the past. It can also lead to discovery of constraints that could have been missed if requirements were drafted from scratch. Furthermore, it can also facilitate mutual learning and business domain knowledge sharing between business and IT stakeholders.

2.12 Requirements Management

Requirements change during the SDLC, especially when agile SDLC methodologies are used in rapidly changing business domains, like the financial services. Managing requirements volatility and keeping stakeholders informed when there are changes is not a trivial task. It is therefore necessary that an organization has well-tested and proven requirements management procedures.

Requirements management involves activities geared towards effectively managing information related to requirements and preserving the integrity of the information through the lifecycle of the solution for which the requirements are intended. This implies that requirements management is continuous throughout the project lifecycle. The effective control of information entails planning, monitoring, analysing, communicating and controlling of information related to the requirements. Tracing the relationship among requirements is also within the scope of requirements management. Requirements management also ensures high level of quality and value of the existing requirements (2014, Schwaber and Sterpe, 2007, Hood et al., 2007).

Requirements management activities from the perspective of BIA are reviewed in this section. Potential issues that can arise when an organization's requirements management processes are weak or non-existent are highlighted. The impact this can have on business-IT alignment is articulated. The discussion forms the background for the section on requirements engineering and management tools that follows.

2.12.1 Requirements Management Activities

The activities involved in requirements management include requirements identification, cross referencing, origin and ownership, change control, and configuration management (Paul et al., 2010). Of the benefits of requirements management, communication and traceability stand out. The diverse attributes related to requirements that can be traced for effective control of a project such as architecture, design, interface, feature, tests, code highlight the essence of a structured and tool supported approach to requirements management. Traceability is required in domains such as financial services to satisfy audit, compliance, or regulatory stipulations.

A PMI study (2014), reveals that 47% of unsuccessful projects fail to meet goals due to poor requirements. At the core of this high failure rate, is the lack of effective requirements change communication. This lends support to assertions made in this thesis on the importance of communication and collaboration between project stakeholders. The study also shows that low requirements management maturity is widespread. It reports that the cause and effect of poor requirements is worse for low-performing organizations and costs them nearly ten cents for every dollar spent in IT investments. It can therefore be argued that improving the maturity of requirements management can contribute to improvements in business-IT alignment.

2.12.2 Alignment Issues Relating to RM

The processes that make up a typical RM programme in an organization include traceability, baselines, change management, and change notification. The efficacy or deficiency of these processes, as the case may be, can have significant impact on business-IT alignment. The constant changes in business needs, business priorities, context, technologies, markets, regulatory demands, require robust and well-structured processes for tracking the changes and notifying stakeholders of these changes. The problems that arise when this is lacking is widely acknowledged (Firesmith, 2007, Jones, 1997). It can range from wasted IT resources expended on development, testing, or documentation for requirements that no longer exist to confusion and unnecessary rework of IT artefacts due to out-dated requirements.

Robust requirements management processes suitable for the peculiar needs of an organization that adheres to RM best practices can be an enabler to business-IT alignment, while poor RM processes can be an inhibitor to business-IT alignment. A business-IT alignment maturity uplift endeavour should therefore contain an action point to assess current RM practices to ensure that changing business needs and priorities are synchronized with IT tasks. This can be supported by specialized tools for requirements engineering and management, which are reviewed in the next section.

2.13 Requirements Engineering and Management Tools

Requirements management (RM) processes and the challenges that organizations face in keeping their requirements current in a rapidly changing business climate have been highlighted. It has been argued that RM processes should be geared towards synchronizing business needs and priorities with IT tasks through effective management of requirement changes. Synchronizing requirements versions when multiple people are working simultaneously is challenging. Some organizations have attempted to use processing tools like word processors for requirements management. This can be useful for small scale projects with few people working on the requirements and in environments with low requirement volatility. The limitations of using word processors become apparent as project scale, stakeholders, and requirements volatility increases. RM tools address these limitations.

RM tools are necessary for ensuring consistency and efficiency in managing requirements change, keeping specifications updated and accessible, notifying stakeholders of changes, supporting multiple user collaboration, and various other functions particularly in large-scale projects. Requirements definition and management tools are commonly used in mission-critical domains such as aerospace, defence, medical device, automotive and telecommunications. These specialized tools are increasingly being adopted in service industries as realization that defective requirements lead to defective solutions takes hold (Schwaber and Sterpe, 2007).

It is necessary at this point to highlight the differences between RM tools and RE tools from a functional perspective. RM tools provide features such as traceability, linking, change notification, collaboration, historic tracking, test case definition, and enabling distributed development (Hoffmann et al., 2004). RE tools offer support for requirements definition, acquisition, specification, grouping and attribution of elicited requirements, support requirements derivation to more detailed levels, and persisting and adjusting of requirements attributes. Despite the differences, there are tools that provide both RE and RM functions.

RE/RM tools either adopt a document-centric or database-centric approach to providing their functionalities. There are differences between both approaches. A document-based approach, as the name implies, functions by importing requirements documents and exporting requirements documents whereas a database-centric approach uses its repository/database as the primary means for presenting, processing and storing requirements. Switching between tools that follow different approaches requires significant training and careful planning. Investment in these tools should also be complemented with investment in user training for optimal utilization of the features these tools offer. This is essential for maximizing ROI in these tools.

2.13.1 Comparative Analysis of RE/RM Tools

There are numerous commercially available RE/RM tools. The commonly used ones include DOORS, RequisitePro, Caliber RM, Cradle, Visure IrQa, and MKS Integrity. Factors to consider in selecting these tools include cost, features, integration with word processors, product domain, project and organization size, reuse and the process maturity (Hoffmann et al., 2004). Analysis of these tools are based on these factors (Carrillo de Gea et al., 2011, Schwaber and Sterpe, 2007, Beatty et al., 2011).

An up to date comparative analysis of these tools by metrics relevant to business and IT priorities synchronization, change notification, collaboration, reuse, ontology support, glossary, query, and mining, is however lacking. To address this gap in literature, a comparative analysis of a selection of commonly used tools was performed and is presented in Table 2.10. It is intended to highlight gaps in the market that tools like REFINTO (proposed in this thesis) can fill, as a useful resource for practitioners making investment decisions on the appropriate RE/RM tools to invest in, and a starting point for academics analysing RE/RM tools.

Table 2.10: Requirements Engineering and Management Tools/Features

Criteria	Blueprint	Cognition Cockpit	Caliber RM	Cradle	DOORS	InteGREAT	Visure (IrqA)	MKS Integrity	PACE	Polarion	Prosareq RM	QPack	RaQuest	Reqfify	Requsite Pro	TopTeam
Definition	✓	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
Specification	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
Elicitation	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Modelling	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✓	✗	✗	✓
Verification and Validation	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Baseline and Versioning	✓	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓
Identification	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗
Change Management	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓
Collaboration	✓	✗	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗
Configuration Management	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗
Documentation Support	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗
Knowledge Support	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Domain Specific	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
Glossary	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Historic Reporting	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Relationship Tracking	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗
Ontology Support	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Integration with other tools	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓
Query and Mining	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Real-time Change Notification	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✗
Repository/Database	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓
Matching and Discovery	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Microsoft Office Integration	✗	✓	✗	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗	✓	✓	✗
Reuse	✓	✗	✗	✗	✓	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗
Role-based Workflow	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
Security	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Test Coverage	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✓	✓
Traceability	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓
Web Interface	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗

2.13.2 Alignment Issues Relating to RE/RM Tools

The case for the use of specialized tools for requirements engineering and management for addressing the challenges that RE and RM activities present especially in rapidly changing business setting has been made in this chapter. However, these tools are most effective when used complementary to existing robust requirements engineering and management processes in an organization. Improper use of these tools or under-utilization of its features can inhibit business-IT alignment and ROI. To avoid this, proper planning and adequate user training is essential. When used properly, these RE/RM tools can facilitate quicker elicitation of high quality requirements, requirements reuse, collaboration, management and communication of requirements changes, and can therefore be an enabler to sustainable business-IT alignment.

2.14 Summary and Discussion

The objectives of this chapter were to build on the foundation laid in chapter one and provide a literature review of the main concepts, trends, and relationship between business-IT alignment and requirements engineering. These concepts are the building blocks on which the thesis is built. The highlights, arguments, and findings made in this chapter are summarized here.

Firstly, the definition of BIA from the different perspectives from which the concept is approached has been presented. Secondly, the main approaches to BIA cited extant literature namely - architecture, governance and communication have been reviewed. Thirdly, the levels at which BIA can be observed, corresponding to the levels of an organisation - strategic, operational and tactical, have been reviewed. Frameworks applied to BIA measurement such as Balanced Score Card (BSC), ITIL, and COBIT have been highlighted. Fourthly, the antecedents to BIA from existing literature are presented. The enormity of the complexities of attaining and sustaining BIA in organizations operating in fast paced and dynamic markets has been highlighted. These discussions form the basis of the proposed means of actualizing the BIA concept in practice.

Requirements engineering has a role throughout the whole software development life cycle, from initial requirements acquisition, to specification, design, implementation, testing and maintenance. Elicited requirements should be clear, complete, and unambiguous before being passed on to the next stage in the SDLC. The process of producing high quality requirements presents opportunities to improve business-IT alignment maturity. Frequent requirement changes, a common occurrence in dynamic and fast moving industries such as the financial services, presents significant challenges which can have adverse impact on BIA. Similarly it presents an opportunity to drive sustainable BIA. This strengthens the argument for rigorous and structured processes requirements management, complemented by appropriate use of RE/RM tools.

The use of these RE/RM tools can be an enabler or inhibitor to business-IT alignment depending on the adequacy of processes and procedures in place that the tools complement. The increasing usage of RE/RM tools in service industries (Carrillo de Gea et al., 2011, Schwaber and Sterpe, 2007, Beatty et al., 2011) following the trend in complex industries such as automobile, defence was also highlighted.

This chapter has makes contributions to literature on business-IT alignment, requirements engineering and requirements management by highlighting how its processes and activities can be leveraged for sustainable business-IT alignment. Although the link between BIA and RE is identified in extant literature, suggestions on how to operationalize and measure this relationship are lacking. This gap in extant literature is addressed in this study. Furthermore, a comparative analysis of RM/RE tools with emphasis on business-IT alignment is presented. The use of intelligent, knowledge-based RE/RM tools has the potential of furthering the advancement of sustainable business-IT alignment by leveraging requirements reuse and patterns. This is considered in the next chapter.

Chapter 3 Knowledge-based Approaches to RE and BIA

3.1 Introduction

In chapter two, the business-IT alignment concept was introduced. The discussion highlighted the problems that can arise when business and IT processes and resources are not aligned. It was argued that the problems of business-IT misalignment are exacerbated when there are communication, knowledge, and language gaps between business and IT stakeholders especially at the tactical and operational levels of the enterprise. It was also highlighted that communication between business and IT stakeholders is intense during requirements engineering (pre-project) activities. This interaction continues through the project implementation (intra-project) and to the deployment (post-project) stage. In emerging software development methodologies such as agile and iterative software development, the quality of this interaction is critical to project success, adjudged by how the IT artefacts meets business expectations within time and budget constraints.

It was also highlighted that requirements in a domain tend to be recurrent and patterns can be identified. These patterns are useful for reuse, knowledge sharing, and learning. The process of capturing requirements for reuse involves some form of knowledge engineering. Based on these linkages, the application of knowledge-based requirements engineering frameworks to drive sustainable business-IT alignment at tactical and operational levels is explored in this chapter. This is achieved by capturing knowledge in the form of requirements and assets for supporting requirements engineering activities such as elicitation and management, facilitating communication between business and IT stakeholders, and learning in the domain. The objective is to investigate if this approach can lead to a practical means of attaining and sustaining BIA maturity.

In this last literature review chapter, concepts of domain engineering, knowledge engineering, and ontological engineering are considered from the perspective of requirements engineering and business-IT alignment. It forms part of the building blocks for chapter five which covers the REFINTO framework and the data analysis and interpretation chapters.

3.2 Domain Engineering for Reuse

Software systems can be classified according to the business function they support such as banking, airline reservation, medical records, order processing, inventory management, etc. Similarly, parts of the software can be classified based on functionality, such as authentication, logging, messaging, etc. Software in the same business domain share many common characteristics as they do requirements. An organization can therefore design for reuse, and reuse components in existing applications and domain knowledge acquired in past development projects for subsequent development projects.

This practice can be beneficial from a cost perspective through reduction of development time and from a learning perspective by enabling learning and sharing domain knowledge. Domain Engineering (DE) is a systematic approach to achieving this goal (Broy, 2013, Czarnecki and Eisenecker, 2000). This indicates that there is an opportunity to apply knowledge based concepts to guide the RE process, requirements patterns, and requirements reuse. Applications of domain engineering in requirements engineering have been explored in extant literature (Sutcliffe and Maiden, 1998, Broy, 2013, Clark and Barn, 2013). An extension of DE concept to business-IT alignment is explored in this chapter.

It can be argued that understanding requirements and domain of application is a fundamental success factor in software development (Dag and Gervasi, 2005) and that understanding the application domain is the first step in requirements elicitation (Zowghi and Coulin, 2005). Based on this, the argument made in this section is that knowledge of the business domain, use of domain models, and reuse of requirements during requirements engineering process can be leveraged for sustainable business-IT alignment.

3.2.1 Domain Engineering Activities

Domain engineering is a continuous and iterative process and its output refined as the domain evolves (Harsu, 2002). There are three activities in domain engineering namely, domain analysis, domain design, and domain implementation as shown in Figure 3.1 (Sodhi and Sodhi, 1999). These are alternatively referred to as the *auxiliary* stages of domain development made of five activities namely, domain (knowledge) acquisition, domain (knowledge) analysis and concept formation, domain (knowledge) verification, domain (knowledge) validation, and domain theory formation.

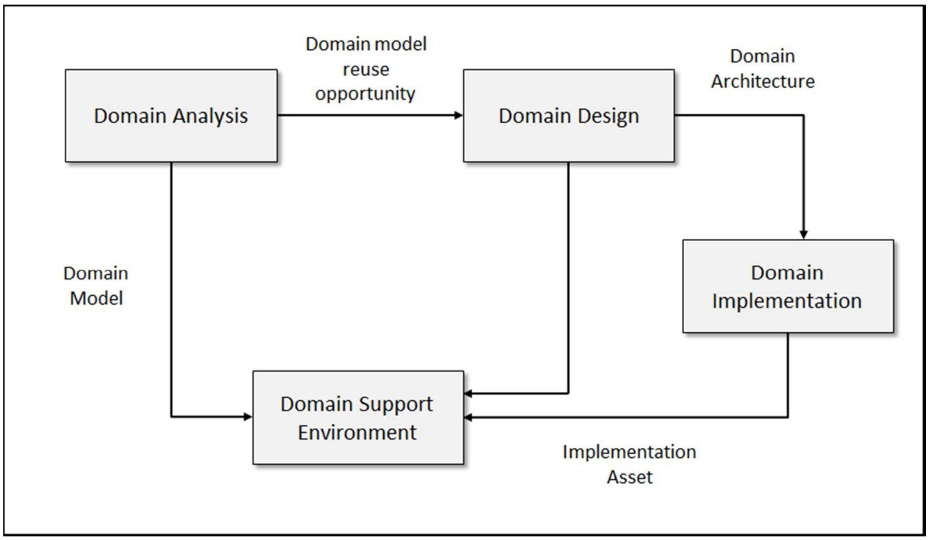

Figure 3.1: Domain Engineering Activities (Sodhi and Sodhi, 1999)

Domain analysis is the first stage of domain engineering. It involves identifying a domain and capturing its ontology (Reinhartz-Berger et al., 2005). Domain analysis activities can be grouped into three categories, namely, building a domain demand range and defining the domain demand boundary, building domain object-oriented analysis model, and confirming domain terms dictionary/lexicon (Meng et al., 2007). The output of domain analysis is a domain model that is well defined and packaged ready for reuse. Domain analysis is an effective means of developing component-based software that maximizes code reuse, minimizes code duplication, and enhances software quality in a short timeframe.

The second stage of domain engineering, domain design, takes a domain model as input and produces a generic design (Alana and Rodriguez, 2007). The domain design stage is subsumed into domain analysis (Weiss and Lai, 1999, Harsu, 2002). Domain design involves gathering information from existing documentations, domain experts, or reuse analyses. Domain design also facilitates building a reference architecture that can be adapted to requirements of future applications. Common rules for the development of specific applications based on a reference architecture are discussed in (Böckle et al., 2005).

Domain implementation takes as input, design models and generic architectures designed in the domain analysis stage and produces reusable assets as output (Alana and Rodriguez, 2007). Domain implementation involves documentation and implementing domain-specific languages and generators. The main aim of domain engineering is therefore to produce reusable assets. At the end of the domain engineering lifecycle, the products are components, feature models, analysis and design models, architectures, patterns, frameworks, domain-specific languages, production plans, and generators (Sodhi and Sodhi, 1999, Caplinskas et al., 2003, Alana and Rodriguez, 2007).

3.2.2 Application Engineering for Reuse

Whereas domain engineering is engineering for reuse, application engineering is reuse of engineering. Application engineering uses the products of domain engineering to produce artefacts that can be reused in the domain. It involves producing a single application by reusing outputs from domain engineering. Application engineering has three stages, similar to domain engineering as depicted in Figure 3.2. These are requirements analysis (application analysis), application design, and application implementation. It also involves customization, which resolves variability according to the application needs (Bryant et al., 2010, Caplinskas et al., 2003, Reinhartz-Berger et al., 2005, Böckle et al., 2005).

Collectively, domain engineering and application engineering are referred to as domain-specific software engineering (DSSE). The three key factors of DSSE are the domain, technology, and business. The domain constrains the problem space and focuses development whereas technology adopts a variety of technological solutions such as patterns, architectures, tools and legacy systems on the domain. Business needs drive the use of DSSE to minimize cost through assets reuse while increasing market share by developing many related applications for variety of business users (Bryant et al., 2010, Caplinskas et al., 2003).

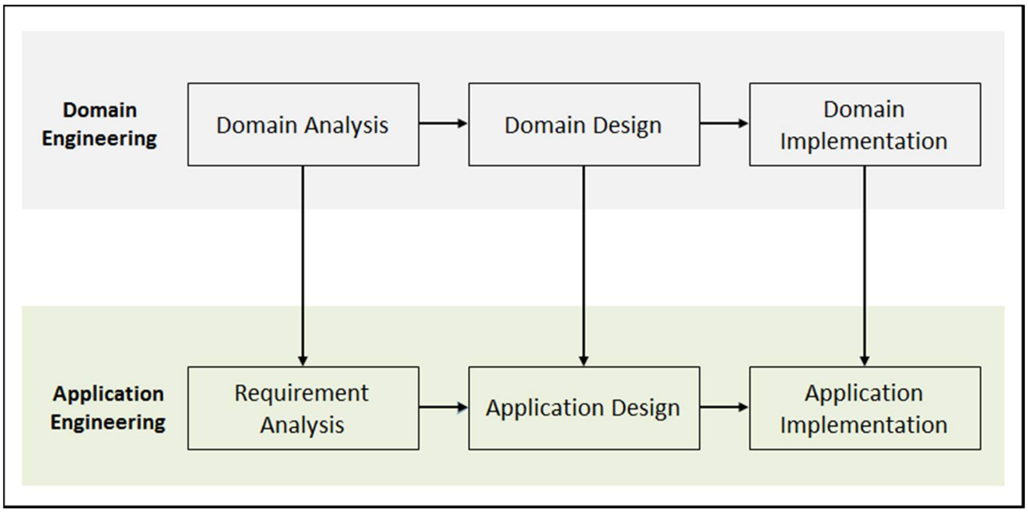

Figure 3.2: Domain Specific Software Engineering

3.2.3 Domain Engineering and Implications for RE

In chapter two, requirements engineering, requirements reuse, and requirements patterns were explored. It has been argued in this chapter that these concepts are relevant to domain engineering. The significance of domain engineering to requirements engineering in activities such as capturing domain knowledge, building models from the domain model for the purpose of requirements, and artefact reuse is explored in this section.

Although research into the use of knowledge-based approaches to addressing business-IT alignment has also been suggested in extant literature, unexplored perspectives to applying knowledge-based requirements engineering to business-IT alignment still remain. The ontology-based requirements engineering framework and supporting tool for addressing business-IT alignment that are proposed, developed, demonstrated, and validated in this study contributes to addressing this gap.

3.3 Knowledge-based Systems and RE

Extant literature on requirements engineering explore the application of artificial intelligence (AI), Knowledge-based systems (KBS), and other knowledge engineering concepts for requirements capture, modelling, and validation. KBSs facilitate the reuse of knowledge about typical information system applications and domains during such activities (Loucopoulos and Champion, 1989, Loucopoulos and Champion, 1988, Gibson and Saeedi, 1995, Kendal and Creen, 2007, Nguyen et al., 2014, Gibson and Conheeney, 1995, Maalej and Thurimella, 2013). This is extended to business-IT alignment in this section.

In chapter two, requirements engineering activities were described as complex, knowledge, resource, and communication intensive activities. This section focuses on knowledge engineering, knowledge-based systems, and their applications to requirements engineering. The objective is to explore the application of knowledge engineering to address software engineering and requirements engineering problems, with emphasis on business-IT alignment.

3.3.1 Knowledge-based System Structures and Activities

Knowledge-based systems or expert systems, use human knowledge to solve problems that normally would require human intelligence to solve (Akerkar and Sajja, 2010, Alavi and Leidner, 1999, Tripathi, 2011). Typical KBSs have means of acquiring, representing, and persisting acquired knowledge in a repository. They also have a reasoning facility (inference engine) and optionally an explanation model, as depicted in Figure 3.3.

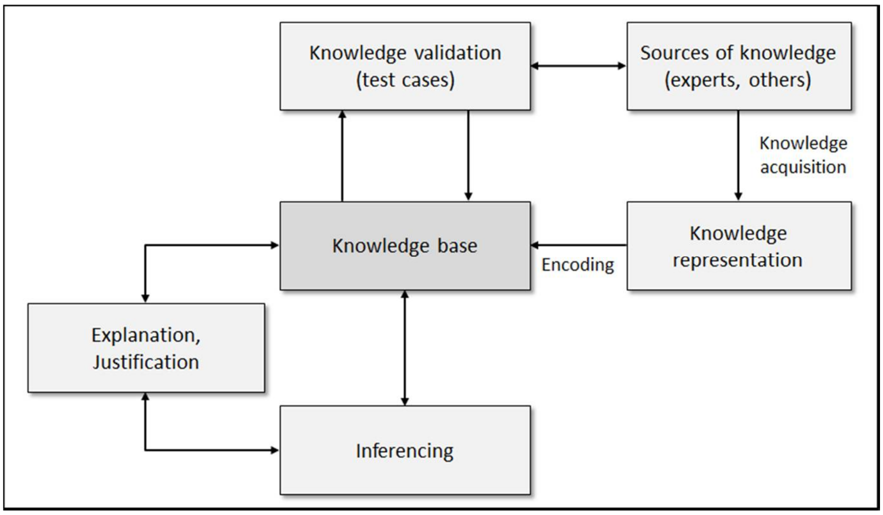

Figure 3.3: Typical Knowledge-based System (Turban and Aronson, 2001)

Similar to requirements elicitation, knowledge acquisition involves extracting knowledge from sources such as domain experts, documents, databases or the web and transferring these to a knowledge repository or to an inference engine for the purpose of reasoning on them. Knowledge acquisition methods can be broadly grouped into manual, semi-automatic (expert-driven), or automatic (computer-aided induction driven). Knowledge acquisition methods include decision trees, data mining, interviews, questionnaires, record reviews, and observation. The shortcoming of these methods is their ineffectiveness in capturing tacit knowledge, which is stored in the subconscious minds of domain experts and reflected in the mental models, insights, values, and actions of the experts. Concept sorting, concept mapping, and protocol analysis have been found to be useful for addressing the challenges of tacit knowledge extraction (Akerkar and Sajja, 2010, Kendal and Creen, 2007, Turban and Aronson, 2001).

Knowledge validation and verification involves testing and refining acquired knowledge to ensure its quality is acceptable and its outputs match the output of human domain experts given the same input data. The challenges of knowledge validation include where knowledge is located and how it can be validated. During knowledge modelling, knowledge validation can take different aspects namely intrinsic validation of the expertise model, building of the KBS according to the expertise model, validation of the KBS according to the expertise model, and questioning the expertise model (Batarseh and Gonzalez, 2013, Jiang et al., 2008, Djelouah et al., 2002, Gonzalez and Barr, 2000, Antoniou et al., 1998, Vicat et al., 1995, Gupta, 1993).

Knowledge Representation (KR) serves the functions of representing acquired knowledge for modelling knowledge, knowledge exchange, storage, and reuse. Natural language representation of knowledge representation has shortcomings namely non-uniformity, ambiguity, and complexities to achieve accurate semantic and syntactic expression. As such, other representation types are used in KBSs such as logical representation, semantic networks, production rules, and frames (Akerkar and Sajja, 2010, Kendal and Creen, 2007).

The knowledge base is a critical piece of a typical KBS setup. Knowledge acquired from experts or other sources has to be persisted to some sort of repository and be readily retrievable for reasoning/inference. Knowledge is stored in the knowledge base in representation formats such as semantic nets, frames, production rules discussed in the previous subsection. Therefore, the knowledge base contains a collection of rules that are used for the problem solving tasks (Turban and Aronson, 2001).

The knowledge reasoning module (inference engine) is the mechanism for reasoning on knowledge. It involves the design of software to facilitate making inferences based on stored knowledge and specifics of a problem and then provide advice to non-expert users. Knowledge reasoning can follow strategies depending on the control direction. This can either be based on data (forward chaining) or goal (backward chaining) (Akerkar and Sajja, 2010, Brachman and Levesque, 2004, Kendal and Creen, 2007, Turban and Aronson, 2001). Reasoning can also be based on rules or cases.

The explanation and justification stage of the knowledge engineering lifecycle involves the design and building of an explanation capability within the KBS to provide answers and reasoning for inputs requested and justification for the conclusions the KBS makes (Turban and Aronson, 2001). It refers to the ability of the KBS/ES to explain how and why it arrived at a certain conclusion. The explanations of a systems conclusion can be as important as the conclusions (Chandrasekaran and Swartout, 1991).

3.3.2 Application of KBS in Requirements Engineering

KBSs are relevant to requirements engineering with respect to requirements elicitation, reuse, and management. Applications of KBSs can be found in various domains such as financial services, human resources management, operations and production, marketing. In financial services, KBSs are applied in processes such as underwriting, claims processing, reserving, auditing, real-time financial applications that require timely transaction processing, and responding to real-time events. In high-frequency trading, KBSs use historic data analysis, real-time data feeds, and advanced algorithms to make high-volume and complex trading decisions in micro-seconds.

Other applications of KBS in financial services include portfolio management, risk management, financial statement analysis, portfolio analysis, credit scoring, stock selection, and data analytics (Leinweber, 1988, Laffey et al., 1988, Matsatsinis et al., 1997, Zopounidis et al., 1997, Shiue et al., 2008, McGowan, 2010, Nakashima et al., 2005). In human resources, KBSs have long been used in training and education (Devedzic, 2004, Wilson and Welsh, 1986). KBSs are also applied in marketing for functions such as strategic market planning (Borch and Hartvigse, 1991, Liberatore and Stylianou, 1994). The applications of knowledge-based systems specifically to requirements engineering include for activities like requirements elicitation, requirements reuse and addressing requirements volatility, and are explored further.

There are parallels between knowledge acquisition and requirements elicitation (Shaw and Gaines, 1996). KBSs are applied to requirements elicitation in various forms such as ontology-based frameworks. An example of the use of KBSs in requirements elicitation is the knowledge based requirements elicitation tool (KBRET) (Gomaa et al., 1996), an interactive and domain-agnostic requirements elicitation tool. KBRET is used to support the development of domain models and specifications generated from the domain models (Gomaa et al., 1996).

More recently, a framework with a support tool for addressing requirements inconsistencies, called knowledge-based RE framework (KBRE) (Nguyen et al., 2014) has been proposed. KBRE is a goal-directed RE process that applies domain knowledge and semantics of requirements for requirements elicitation, elaboration, and detection of inconsistencies and other related requirements problems. It uses description logic *SROIQ* as the fundamental logical system for analysis of and reasoning about requirements. The main focus of KBRE is identifying inconsistencies between requirements as well as redundancies and overlaps of requirements. Manchester OWL Syntax (MOS) (Horridge et al., 2006), is used as the requirements specification language to facilitate creation and maintenance of ontologies for storing the domain knowledge and semantics of requirements.

Reuse is essentially based on using or modifying past experiences and assets to solve new problems. This is similar to a case-based reasoning activity. KBSs are useful for requirements reuse. Tools such as KBRET are useful towards software requirements and reuse. The benefits of requirements reuse using knowledge engineering constructs include lower development costs, higher productivity, reduced development time, lower training costs, easier software maintenance, higher requirements quality, lower project risk, improved software system interoperability, easier mobility of personnel, tools, and methods between project (Homod and Rine, 1999, McClure, 1997).

KBSs are also useful for managing requirements volatility. This is driven by the need to mitigate the risks associated with requirements volatility such as imprecision, multiplicity, and conflicts in quality goals (Palmer and Myers, 1988).

KBSs are proposed as a means to address these risks through requirements verification and validation. Palmer and Myers make a case for using KBSs as a means of managing requirement volatility, mitigating risks, ensuring high quality requirements, and reducing errors in software. This provides support for the arguments made in this thesis.

3.4 Ontological Engineering for KBS

Ontologies can be defined as hierarchical sets of terms that describe an arbitrary domain or a specification of a conceptualization which enables knowledge sharing and reuse (Gruber, 1993, Gomez-Perez et al., 2004, Guarino and Welty, 2000). Ontologies make domain knowledge explicit and provide a medium for sharing semantics, domain models, and improvement in communication as is the case in the semantic web.

Ontologies are used in knowledge engineering as a means to conceptualize, formalize, make domain knowledge machine-interpretable, and support navigation, search, and retrieval of knowledge (Gavrilova, 2010). Ontologies are also thought to hold great promise for software reuse (Falbo et al., 2002), a theme central to this thesis. The applications of ontology to software engineering, specifically requirements engineering is explored in this section.

3.4.1 Application of Ontology in Software Engineering

Ontologies have wide applicability throughout the software development life cycle (SDLC) especially at the analysis and design stages which involves requirements engineering. Ontologies facilitate the formal specification of the domain thereby making domain models first order citizens in software engineering tasks. Ontologies also provide support for modularisation, distribution, reuse, and integration of software components (Happel and Seedorf, 2006).

There are other useful applications of ontologies such as neutral authoring (authoring ontology in one language and translation to various formats to target multiple applications), specification, providing common access to information, ontology-based search (Falbo et al., 2002), ensuring consistency and correctness of domain knowledge by constraining the content of information, supporting inference to derive additional knowledge from a set of facts, and creating libraries of interchangeable and reusable models (Aldea et al., 2003).

Specific applications of ontologies in knowledge-intensive business processes in financial services can be found in business functions such as corporate action processing, insurance underwriting, pension entitlement calculation, sales commission calculation, pension benefit calculation, exception management (Bhat et al., 2007).

3.4.2 Application of Ontology in Requirements Engineering

Ontologies have been found to be useful in requirements engineering. It has been argued that interaction between business and IT stakeholders is intense during the requirements engineering stage of software development projects. Knowledge and language gaps between stakeholders during this process can lead to problems with requirements including imprecision, ambiguity and duplication, and subsequently software quality problems. This can be addressed with the application of ontologies. Ontologies can be useful towards knowledge representation and facilitating knowledge sharing especially in industries with specialized vocabulary (Siegemund et al., 2011, Omoronyia et al., 2010, Yang et al., 2008, Dobson and Sawyer, 2006, Kaiya and Saeki, 2006).

Ontologies are also useful for requirements specification documentation (Castañeda et al., 2014, Happel and Seedorf, 2006, Decker et al., 2005, Mayank et al., 2004) and formal representation of requirements knowledge (Castañeda et al., 2014, Happel and Seedorf, 2006, Lin et al., 1996, Wouters et al., 2000). The expressiveness of ontologies can be adapted to cover semi-formal, structured and formal representation (Happel and Seedorf, 2006, Wouters et al., 2000).

Furthermore, the concepts, relations and axioms of a domain ontology can be used to provide a list of suggestions, which a requirements engineer can use to define requirements as proposed in a semantic guidance system (Farfeleder et al., 2011).

Domain models for a problem domain can be represented using ontologies such as glossaries and conceptual diagrams like UML. The advantages ontologies offer in this context include being well suited for evolutionary specification of requirements and domain knowledge (Happel and Seedorf, 2006, Wouters et al., 2000), supporting requirements management and traceability, automated validation and consistency checking which offer logical formalism, and formal specification (Happel and Seedorf, 2006, Mayank et al., 2004, Lin et al., 1996). In extant literature, it is argued that ontologies should be sub-products of requirements engineering (Gašević et al., 2009, Breitman and Sampaio do Prado Leite, 2003).

In general, ontologies are well suited for reconciling gaps in knowledge and common understanding among stakeholders during the requirement elicitation stage. This subsequently leads to improvements in the quality of elicited requirements (Omoronyia et al., 2010).

3.4.3 Application of Ontology for Component Reuse

The benefits of software reuse in the form of requirements and artefact reuse, such as shorter time-to-market, reduction in duplication of effort, cost savings, enhanced productivity, have been highlighted. Based on these benefits, it has been argued that business-IT alignment can be enabled by leveraging requirements patterns and reuse during the requirements engineering. Ontologies are useful for identifying requirements patterns. For example, a means of transforming use case descriptions expressed in a controlled natural language into ontology has been proposed (Couto et al., 2014). Ontologies can be used for the search, matching, and reuse of components during the implementation stage of the SDLC. However, this should be accompanied by a robust means of managing the challenges that ontologies reuse presents, especially when they are owned by third parties.

There are challenges in retrieving details of components from a reuse repository that are based on syntactical keyword search such as low recall and precision (Mili et al., 1998, Happel and Seedorf, 2006). Ontologies offer the ability to describe component functionality in knowledge representation formalisms that support storage of semantic description of components in a knowledgebase and more powerful querying. Ontologies also make it possible to join information usually stored in isolated component descriptions and providing background knowledge (Happel and Seedorf, 2006).

3.4.4 Application of Ontology for Software Development

Ontologies are useful for coding support by allowing developers to annotate application programming interface (API) elements with an unambiguous concept. Furthermore, ontologies are useful for code documentation by providing a unified representation for problem domain and source code thereby enabling easier cross-references among these information spheres. Ontologies also provide a mechanism to capture knowledge about the problem domain such as library dependencies that can be reused for other purposes.

Enterprises that face changing business environments which require flexibly in their business rules such as those in the financial services domain, can benefit from applying ontologies. Rather than hard-coding business rules in source code, declarative specification, user-friendly, and easily editable representation of the rapidly changing knowledge can be implemented instead (Happel and Seedorf, 2006).

3.4.5 Ontology Representation Languages and Tools

Ontology representation languages are necessary to facilitate reuse of formally represented knowledge among AI systems (Gruber, 1993). The commonly used ontology representation languages are Ontolingua (Gruber, 1992), RDF(S) (Resource Description Framework Schema), OWL (DAML+OIL).

The need for tools to support the construction of ontologies continues to increase as semantic-aware applications become even more prevalent in industry (Croisier, 2012, Léger et al., 2009). Commonly used tools include OntoEdit (Sure et al., 2009, Sure et al., 2004), WebODE (Arpírez et al., 2003, Arpírez et al., 2001), Protégé[2], OE:Ontology editor in Hozo (Kozaki et al., 2007, Sunagawa et al., 2004). These tools provide graphic interfaces for creation and maintenance of ontology in formats such as XML, OWL (DAML+OIL (back ends)), and OIL (tab) or RDF.

Protégé is widely considered the leading ontological engineering tool. It is built on complex software architecture and is easily extensible through plug-ins. It supports interfaces to other knowledge-based tools like Jess (Eriksson, 2003), a rule engine and scripting environment written in Java. It has a MOF-compatible meta-model that resides in the same meta-level as UML and incorporates the Open Knowledge-Base Connectivity (OKBC) knowledge model (Gašević et al., 2009, Gennari et al., 2003). Protégé is used for the design and implementation of the REFINTO framework domain ontologies.

3.4.6 Ontology-based Systems

The categorization of ontology-based systems depends on how ontologies are used (in software or infrastructure) and at what point (at run-time or development time) (Happel and Seedorf, 2006). The categorizations are ontology-based architecture (OBA), ontology-driven development (ODD), ontology-enabled development (OED), ontology-enabled architecture (OEA) as depicted in Figure 3.4.

Ontology-based architecture systems are basically systems in which ontology as first class citizens in applications and are used at run-time whereas ontology-enabled architecture systems use ontology to provide infrastructure support for applications at run time. Systems that use ontology at development time for problem domain description referred to as like ontology-driven development systems whereas ontology-enabled development systems use ontology to provide

[2] Protégé is a free ontology editor available at: http://protege.stanford.edu/

infrastructure support to developers such as component search at development time (Happel and Seedorf, 2006).

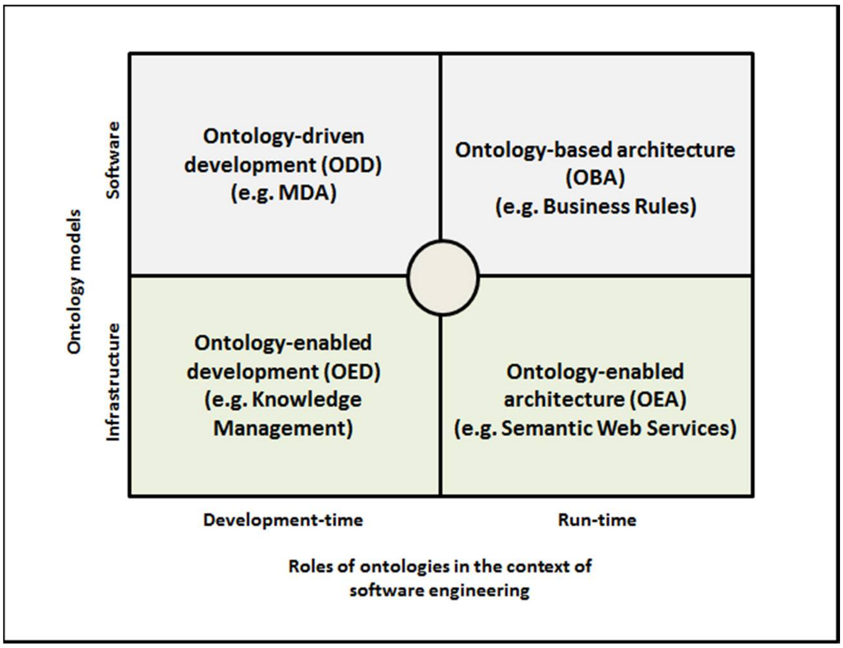

Figure 3.4: Ontologies in Software Engineering (Happel and Seedorf, 2006)

3.5 Ontology-based RE Framework for BIA

There is support for the application of ontology-based requirements engineering frameworks for various purposes in specialized industries in extant literature (Siegemund et al., 2011, Omoronyia et al., 2010, Yang et al., 2008, Dobson and Sawyer, 2006, Kaiya and Saeki, 2006). These include aerospace (Kossmann et al., 2008), engineering design (Lin et al., 1996), engineering decision support (Arndt and Klein, 2002), and supply chain management (Chandra and Tumanyan, 2004). Ontology-based requirements engineering frameworks and tools proposed include OntoREM (Kossmann and Odeh, 2010), OntoRAT (Al-Hroub et al., 2009), and OntoQA (Tartir et al., 2005).

Within ontology-based requirements engineering research, focus has been on aspects such as requirements elicitation (Farfeleder et al., 2011, Shibaoka et al., 2007, Lee and Zhao, 2006), conflict analysis (Liu, 2010), requirements quality

(Heidari et al., 2013, Li et al., 2010), requirements traceability (Assawamekin et al., 2010), requirements formalization (Babkin and Potapova, 2010), and security requirements (Chikh et al., 2011, Tsoumas et al., 2005).

The potential benefits of using ontology-based requirements engineering frameworks for business-IT alignment maturity improvement and sustainability has not however been sufficiently explored in extant literature. This study makes contributions towards filling this gap. One framework that suggests the use of ontology-based requirements engineering for business-IT alignment is the B-SCP framework (Bleistein et al., 2006). It proposes an approach that provides support for BIA through Requirements Engineering. Its objective is to ensure that system requirements of organizational IT is in alignment with and provides support for competitive strategy. It combines requirements engineering methods for formalizing and reasoning about software with analytical frameworks for competitive business strategy. This implies that B-SCP focuses at the strategic level of the enterprise and is therefore susceptible to the limitations of frameworks and tools that operate at this level in regards to BIA that have been highlighted in this thesis.

To evaluate and validate the B-SCP framework, two worked examples using data from completed, strategic IT projects of varying scale and complexity, occurring in different industrial applications is used. The number of projects the framework is evaluated and validated against highlights a potential weakness. It also hints of a lack of rigour and limited scenarios that the framework has been validated against. B-SCP is also criticized for its weakness in tracking complex projects (Veres et al., 2010). This indicates potential weaknesses that may inhibit the application of the framework to drive business-IT alignment in specialized industries such as financial services. Furthermore, B-SCP does not define metrics and processes for measuring BIA maturity obtained using it.

The REFINTO framework and tool attempts to address the limitations and shortcomings of B-SCP and other framework focusing solely on the strategic level in achieving sustainable BIA. To achieve this, the REFINTO framework, an ontology-based requirements engineering framework, approaches BIA from the operational and tactical levels. The framework also defines metrics for measuring BIA maturity. The framework is validated against real-life and business-critical projects of significant scale and complexities in a top financial services firm. The REFINTO framework and tool are discussed in chapter five.

3.6 Summary and Implications for REFINTO

This chapter concludes the literature review of the thesis. Concepts and extant literature on knowledge-based requirements engineering from the perspective of sustainable business-IT alignment have been reviewed. The objective being to highlight the potentials that the use of domain knowledge acquired over time and reuse of artefacts can have on sustainable BIA. This is based on the argument that knowledge-based requirements engineering can be useful towards closing the knowledge and language gaps between business and IT stakeholders during the requirements engineering stage of projects as a means of improving business-IT alignment maturity. This is the basis of the REFINTO framework and support tool. The framework and tool are based on requirements engineering, knowledge engineering, domain engineering, and ontological engineering concepts reviewed in this chapter.

Chapter 4 Research Design

4.1 Introduction

The research design for this study is discussed in this chapter. The ontological, epistemological, and methodological assumptions for the study, which influenced the research method adopted for the study are highlighted. After a careful consideration of the philosophical assumptions and the nature of the study, the design research method (DR) (Hevner et al., 2004, Hevner and Chatterjee, 2010) was adopted for the study.

This study was mainly a deductive and exploratory research. A post-positivist and pragmatist stance was taken. The data collection method was through role-based and stage-differentiated surveys. The data analysis method adopted was mainly quantitative. Statistical analysis was done on the data collected and structural equation modeling (SEM) was used to validate the hypotheses and the correlation between the factors/constructs in the study.

The decision points for the various stages of an IS or empirical SE research such as this are illustrated in this study chapter. Also, the justification for selecting the research methodology and data collection and analysis methods are expanded on. The steps, processes and cycles of design research (DR), and how this study met these and the stated criteria for conducting a DR study are demonstrated. This expands on the introductory discussions in chapter one.

4.2 REFINTO Research Design Strategy

The decision on the appropriate research strategy for an IS and empirical SE research can be challenging, as highlighted in literature (Wohlin and Aurum, 2014, Easterbrook et al., 2008, Galliers and Land, 1987). A decision framework proposed for guiding the strategy (Wohlin and Aurum, 2014) is used to articulate the decisions made and strategy adopted for this study. The Wohlin and Aurum framework has three stages in the decision space, namely the strategy, tactical, and operational phases. The decision points in the framework are research outcome, research logic, research purpose, and research approach which fall under the strategy, research process and research methodology under the tactical phase, and data collection methods and data analysis methods under that operational phase. The research design for this study is discussed using the Wohlin and Aurum framework starting with the research questions investigated in this thesis.

4.2.1 Research Questions

Research questions drive the choice of research methodology, data collection and data analysis methods (Wohlin and Aurum, 2014, Chen and Hirschheim, 2004). The research questions of this study are explained in some detail and the approach to answering them is articulated in this section.

4.2.1.1 Main Research Question (MRQ)

The main question of this study was:

Would an ontology-based requirements engineering framework and tool-supported approach applied to agile/RAD applications projects implemented at tactical and operational levels cumulatively contribute to improved business and IT alignment at these levels?

The key to answering this question and the sub questions derived from this main question have been identified in the introduction and literature review chapters. Firstly, the adverse consequences of misalignment between business and IT including lower return on IT investment, losses incurred on IT projects that do not meet business needs, and loss of competitive advantage have been identified. Secondly, different approaches proposed as means of achieving business-IT alignment have been reviewed. Thirdly, the organizational level alignment can be approached namely strategic, operational, and tactical have been highlighted.

Fourthly, the shortcomings and limitations of approaching alignment from the strategic level alignment have been highlighted. Lastly, the gaps in extant literature on practical means of achieving sustainable business-IT alignment with empirical evidence have been highlighted.

It was argued in the introduction and literature review chapters, that the knowledge and language gaps between business and IT stakeholders is an inhibitor to business-IT alignment. This is especially the case at the tactical and operational levels of an organization where execution of business strategies occurs. It has also argued that these requirements engineering stages of software development can be adversely impacted if there are knowledge and language gaps between business and IT stakeholders and that the problems at that stage propagate throughout the lifecycle of the project.

It is argued that the requirements engineering stage presents an opportunity to leverage the interaction and collaboration between business and IT stakeholders together with knowledge-based rigorous and structured processes for requirements elicitation, management, and reuse to drive sustainable business-IT alignment. This is based on the assumption that when taken as a whole, business-IT alignment for individual projects can contribute to organizational (strategic) BIA. This is backed by the micro-foundational perspective to business-IT alignment (Coltman et al., 2015, Karpovsky and Galliers, 2015).

The main question that this study aims to answer was if these arguments are supported by results of empirical analysis of data collected from real projects using the framework and artefact developed as part of this study. The sub-questions derived from this main question contributed to exploring practical means of actualizing, measuring, and sustaining business-IT alignment. These sub-questions are equally interesting and challenging.

4.2.1.2 Sub Research Question 1 (SRQ1)

The interaction between business and IT stakeholders is most intense during requirements engineering (pre-project) stages. In some organizational settings this interaction ends at this stage. The business stakeholders are only engaged again during the testing stage. The first sub question was aimed at investigating how alignment for a project is influenced if the interaction between business and IT stakeholders is maintained through the implementation stage (intra-project) to the post-implementation (post-project) stages. This inspired the first sub-question of the study:

Can this framework be made to support intelligent and on-demand reasoning and decision making during the requirements engineering process, during project implementation and during evaluation after project implementation?

It has been argued that the quality of requirements, methods, process of eliciting managing, and implementing the requirements can influence the quality of IT artefacts developed based on the requirements. Based on this argument, business-IT alignment was measured at the three project stages for each project and the correlation between the alignment scores at the three stages analysed for trends to validate this assumption.

4.2.1.3 Sub Research Question 2 (SRQ2)

The second sub research question was based on the argument that the choice of elicitation method, the SDLC method, processes, structures, and tools or the lack of these, can have an impact on business-IT alignment. The second sub-question of the study therefore was:

Would the framework facilitate a more structured and rigorous requirements engineering process than an ad hoc approach in agile/RAD projects without encumbering the process with the demands of the waterfall development method such as delaying projects till formal requirements documents are signed off?

To investigate this, the framework and the associated artefact implemented were applied to subsets of projects, one using the framework only and the other combined with tool support. The alignment scores achieved in these projects were compared to the alignment scores achieved in projects using the ad hoc processes and another subset using a third-party framework as control.

4.2.1.4 Sub Research Question 3 (SRQ3)

Measurement of alignment maturity is an important aspect of any research effort for understanding business-IT alignment. To gauge the alignment maturity attained in each of the projects in the four portfolios of projects investigated under sub research question two (SRQ 2), metrics related to the antecedents to operational and tactical business-IT alignment backed by theoretical support and of practical relevance had to be identified and operationalized. These metrics are applied at the appropriate stages (pre-project, intra-project, and post-project) in the project lifecycle to objectively and empirically assess alignment for individual projects and for portfolios of projects. This is used to validate the efficacy of the proposed framework for sustainable business-IT alignment. This is the basis for the third research question:

By what metrics and means can BIA be measured at tactical and operational levels and can these be evaluated through real life projects using the proposed framework?

The selected metrics are those that can be readily replicated in other domains and organizations. The metrics are easy for business and IT stakeholders to understand and operationalize.

4.2.1.5 Sub Research Question 4 (SRQ4)

A gap identified in extant literature is that enablers and inhibitors of alignment at the operational and tactical levels have not been sufficiently researched and literature on this is lacking. In contrast, most of extant literature on antecedents to business-IT alignment focuses on strategic level alignment. The objective of the fourth sub research question was to address this gap by identifying and empirically validating the antecedents to business-IT alignment at operational and tactical levels. The fourth sub research question of this study was:

What are the antecedents to BIA and forces that influence BIA at the tactical and operational levels?

To achieve this, the eight hypotheses on antecedents to business-IT alignment at the operational and tactical levels are validated through the application of the metrics related to the antecedents on the four portfolio of projects investigated under sub research question three (SRQ 3).

4.2.2 Research Outcomes

The outcome of a research can either be basic or applied research. In basic research, the researcher seeks to understand phenomena without proffering solutions. The main contribution of basic research is the generation of knowledge. In contrast to basic research, applied research involves providing solutions to the problem after it has been understood by applying knowledge (Wohlin and Aurum, 2014, Nunamaker et al., 1991, Collis and Hussey, 2009). This study was primarily an applied research. It attempted to address the business-IT misalignment problems in practice by applying ontology-based requirements engineering framework, tool, and processes to an otherwise chaotic and ad hoc process.

4.2.3 Research Logic

Research logic relates to the direction a study takes. This can either be from specific to general (inductive) or from general to specific (deductive). In inductive research, the researcher starts with specific observation, detects theoretical patterns and develops some general conclusions or theories. It is basically a theory-building process and follows a bottom-up approach to developing the theory (Wohlin and Aurum, 2014, Bhattacherjee, 2012, Basili, 1993). In contrast, deductive researchers establish hypotheses by using theory. The researcher collects data to confirm or reject the hypotheses following a top-down approach. Deductive researchers use quantitative research as a means of testing theory.

This study primarily followed the deductive research logic by moving from general underlying concepts of requirements engineering, ontological engineering, knowledge engineering, and knowledge management to the specific idea of an ontology-based framework for requirements engineering in the financial services domain.

4.2.4 Research Purpose

Research can generally be classified by its purpose as exploratory, descriptive, explanatory, or evaluation (Collis and Hussey, 2009). In exploratory research, the researcher aims to gain insight about the phenomena. The aim is to provide background information about the phenomena so that descriptive and explanatory researchers can investigate further. Exploratory research can follow either quantitative or qualitative methods. Descriptive research seeks to describe the phenomena. The research questions tend to start with *how* and *what*.

Explanatory research is used when the causal relationship between the factors and constructs that are present in the phenomena under study is to be understood. The research questions tend to start with *why*. Statistical analysis is used to discover and better explain the causal relationships. Evaluation research aims to determine the impact methods, tools or frameworks have on the phenomena under study and may use exploratory, explanatory and descriptive research. This type of research in the context of engineering is referred to as “improving” research (Runeson and Höst, 2009, Wohlin and Aurum, 2014).

This study can be primarily classified as an exploratory research. However, it should be noted that there are aspects of the research that were descriptive and explanatory in nature. For example, this study advocated for improving an otherwise ad hoc requirements engineering process with a structured and disciplined process.

4.2.5 Research Approach

The paradigm followed in a research project is based on its ontological, epistemological and methodological assumptions, which in turn determines the research approach adopted. Empirical software engineering research tend to follow four paradigms - positivist (post-positivist), constructivism (interpretivist), critical research or pragmatism (Easterbrook et al., 2008). This study primarily aligned to a post-positivist and pragmatist stance. This is due to the use of hypotheses derived from concepts and theories in IS and SE as the basis of/assumption in the design and implementation of framework and the artefact.

4.2.6 Research Process

There is support in extant literature for the combination of aspects of quantitative research with qualitative research to complementarily leverage the strengths of each approach. This informs the emergence of the *mixed methods* research processes. This research was mainly quantitative with some qualitative aspects as such can be classified as a mixed methods research. The data collection method discussed later in this chapter was through qualitative means, whereas the data analysis and interpretation was done using quantitative means.

4.2.7 Research Method

The most critical aspect in the decision making process for research design is the choice of the research methods, processes and frameworks. The choice of research method for this study was primarily design research. Design research tends to be explorative in nature. This is because the creation of the artefact is essential before it can be evaluated. Furthermore, design research can use both qualitative and quantitative research approach (Wohlin and Aurum, 2014).

4.2.8 Data Collection Method

Data collection method for a research depends on the research questions (Benbasat et al., 1987). Data collection methods can be quantitative or qualitative data, or as argued in cited literature, a combination of both. Quantitative data collection methods include archival research, surveys, experiments, and simulation (Wohlin et al., 2012, Wohlin and Aurum, 2014). Qualitative data collection methods used in empirical software engineering includes observation and participant observation (Seaman, 1999). The data collection method used in this research was primarily a closed form, 34-item, role-based and stage-differentiated questionnaires described in Appendix C. Archival research was also performed to analyse past project metadata such as requirements, emails, project plans etc.

The questionnaires were tailored to the roles aligned to business and IT for the three defined stages in the project lifecycle (pre-project, intra-project and post-project). The questionnaires were served through the data and evaluation module of the REFINTO framework support tool. This is discussed in more detail in chapter six.

4.2.9 Data Analysis Method

Similar to data collection methods, data analysis methods can be quantitative or qualitative. Quantitative data analysis methods include statistical analysis and mathematical modelling. Quantitative data analysis requires technical and analytical skills. Qualitative data analysis methods include thematic analysis and hermeneutics and may use qualitative data such as text. Some data analysis methods such as grounded theory, which is also considered a research method, can use both quantitative and qualitative data.

The data analysis adopted for this study was quantitative statistical analysis. Structural equation modelling (SEM) (Brown, 2015, Kline, 2005, Ullman, 2006) was used to validate the hypotheses and the correlation between the factors/constructs in the study. SEM, also known as path analysis, is a collection of statistical techniques for the relationships between one or more independent variables or factors which may be discrete or continuous and dependent various, which may also be discrete or continuous to be examined outcomes (Mueller and Hancock, 2008, McDonald and Ho, 2002). The statistical techniques in SEM include confirmatory factor analysis (CFA), exploratory factor analysis, or principal component analysis (PCA). Each of these techniques has specific contexts that they are suitable for. CFA is used when factor structure of a set of observed variables are known a priori based on theoretical research whereas EFA is used when that structure is not known. CFA allows researchers to test hypotheses that a relationship exists between observed variables and underlying latent constructs. CFA uses covariance whereas EFA uses correlation between factors. PCA is an exploratory technique used to examine how factors linearly combine to produce a set of uncorrelated composite outcomes (Suhr, 2006, Mueller and Hancock, 2008).

There are number of commercially available tools that are commonly used in SEM research. The more commonly used tools include LISREL (**LI**near **S**tructural **REL**ationships) (Jöreskog and Sörbom, 1996), EQS (**EQ**uation**S**) (Bentler, 1995), and AMOS (**A**nalysis of **MO**ment **S**tructures) (Arbuckle, 1997), which is built into SPSS. Other tools such as CALIS, MPLUS, and SEPATH are also used.

Reporting SEM research data analysis can be challenging. There are guidelines and best practices recommended (Jackson, 2009, Schreiber et al., 2006). The recommended reporting structure follows the SEM research steps. This structure is followed in chapters six and seven covering the SEM data analyses performed.

4.3 Justification for Research Design Strategy

Deciding between action research (AR) and design research (DR) as a research method can be challenging. Whereas some aspects of this research are clearly design research oriented, others such as initiating change by introducing a structured approach to an otherwise ad hoc process in order to address business-IT alignment, clearly a socio-technical issue in an organizational setting, seemed more suited to action research. The development of an artefact as part of intervention for change in the organization is a design research activity. However, the testing and evaluation of the artefact is an activity in both action research and design research.

Whereas action research is mainly qualitative in nature, the mixed methods approach of design research, due to its roots in engineering and natural sciences was considered more appropriate. The use of four portfolios of projects for validating the metrics, framework, and tool were quantitative in nature. As with mixed methods research process, there is emerging support for the combination of action and design research especially for addressing weakness and leveraging the strengths of both research methods (Jrad et al., 2014). A checklist on how this study satisfied the criteria for using DR is provided in Table 4.1.

Table 4.1: Design Research Guideline

Question	Answer
What is the research question (design requirements)?	The research questions are articulated in chapters one and four
What is the artefact? How is the artefact represented?	The artefact is a framework and a support tool at three defined project stages
What design processes will be used to build the artefact?	The search for a viable solution that can address the business-IT misalignment problems
What, if any, theories support and artefact design and the design process?	Theoretical backings for the hypotheses underpinning the REFINTO framework
What evaluations are performed during the internal design cycles?	Role-based evaluations of the framework and artefact at three defined project stages
What metrics are used to demonstrate artefact utility and improvement over previous artefacts?	Testing of the artefact was introduced at three defined project stages using pre-defined metrics for the stages
What new knowledge is added to the knowledge base and in what form?	The contributions of the research include alignment metrics, framework, models, and published papers

4.4 Summary

In this chapter, the decision points in selecting appropriate research design strategy for this study were elucidated. The justification, challenges and decision process involved in selecting the research method for this study was articulated. The research design strategy and how it maps to DR method was discussed. This chapter forms the foundation on which the next chapters covering the description of the theoretical and hypothetical assumptions for the REFINTO framework, the artefact, the data collection and analysis to validate the artefact is based.

Chapter 5 REFINTO Framework and Support Tool – Concepts, Design and Development

5.1 Introduction

The gaps in extant literature that this study attempts to address have been highlighted in preceding chapters. It includes the lack of practical approaches to business-IT alignment that can be objectively measured. It also includes the lack of metrics for alignment measurement that can be operationalized and are relevant to application in daily operations of an organization. Furthermore, most proposed frameworks and models for business-IT alignment are not validated through application in real life settings. The argument for an ontology-based framework to guide the SDLC and PMLC processes starting at the requirements engineering as a means of attaining and sustaining business-IT alignment has been made.

This chapter concentrates on the REFINTO framework, its components, workflows, and processes. The framework support tool, its architectural design, implementation, and domain ontologies are presented. The use of the framework and tool is demonstrated. A comparative analysis of the REFINTO framework and alternative frameworks is provided. The eight hypotheses that underpin the framework are also discussed.

5.2 REFINTO Framework: Overview

The main working assumption of the REFINTO framework, is that a structured and rigorous approach to RE, SDLC, and PMLC processes can be leveraged to improve business-IT alignment by facilitating closer interaction and collaboration between business and IT stakeholders, especially in knowledge-intensive and rapidly evolving domains such as financial services and when tacit knowledge is involved.

For this approach to be effective in improving business-IT alignment, the knowledge and language gap between business and IT stakeholders has to be narrowed. To achieve this, reuse of knowledge gained from historic projects captured in forms such as requirements, IT artefact, and constraints can be useful in a number of ways. Firstly, it can be a reference point stakeholders can refer to whilst discussing current business needs. Secondly, past constraints can provide early alerts of rules, policies, or limitations that may impact on the realization of new business needs that may have been overlooked. Thirdly, the reuse of requirements and IT artefacts can lead to increased productivity, reduced project duration, and enhancement of existing IT artefacts. Fourthly, it can be a useful medium for learning about the business domain.

The process of matching current business needs to past projects and reusing requirements, constraints and artefacts can be time consuming and error-prone if performed manually. Capturing and persisting requirements, constraints, and metadata of related IT artefacts on a continuous basis requires some form of automation. An ontology-based knowledge system can serve these functions and contribute to minimizing incidences of software development projects that are challenged, scrapped, or end in other unsatisfactory outcomes. These undesirable outcomes are often symptomatic of low business-IT alignment maturity.

The framework was conceptualized to make the requirements engineering process more collaborative, minimize misunderstandings between business and technology stakeholders, and help mitigate risks of negative project outcomes. It is also envisaged to facilitate transparent reporting of project estimates and implementation progress. This enables increased business visibility of IT processes and better IT visibility of business priorities. The framework, depicted in Figure 5.1, assumes the delineating of project management lifecycle into three stages namely pre-project (before the project begins) stage, intra-project (during the project implementation) and post-project (after the project has been completed) stage, referred to throughout the thesis. The activities at each project stage and mapping to SDLC and PMLC activities is provided in Table 5.1.

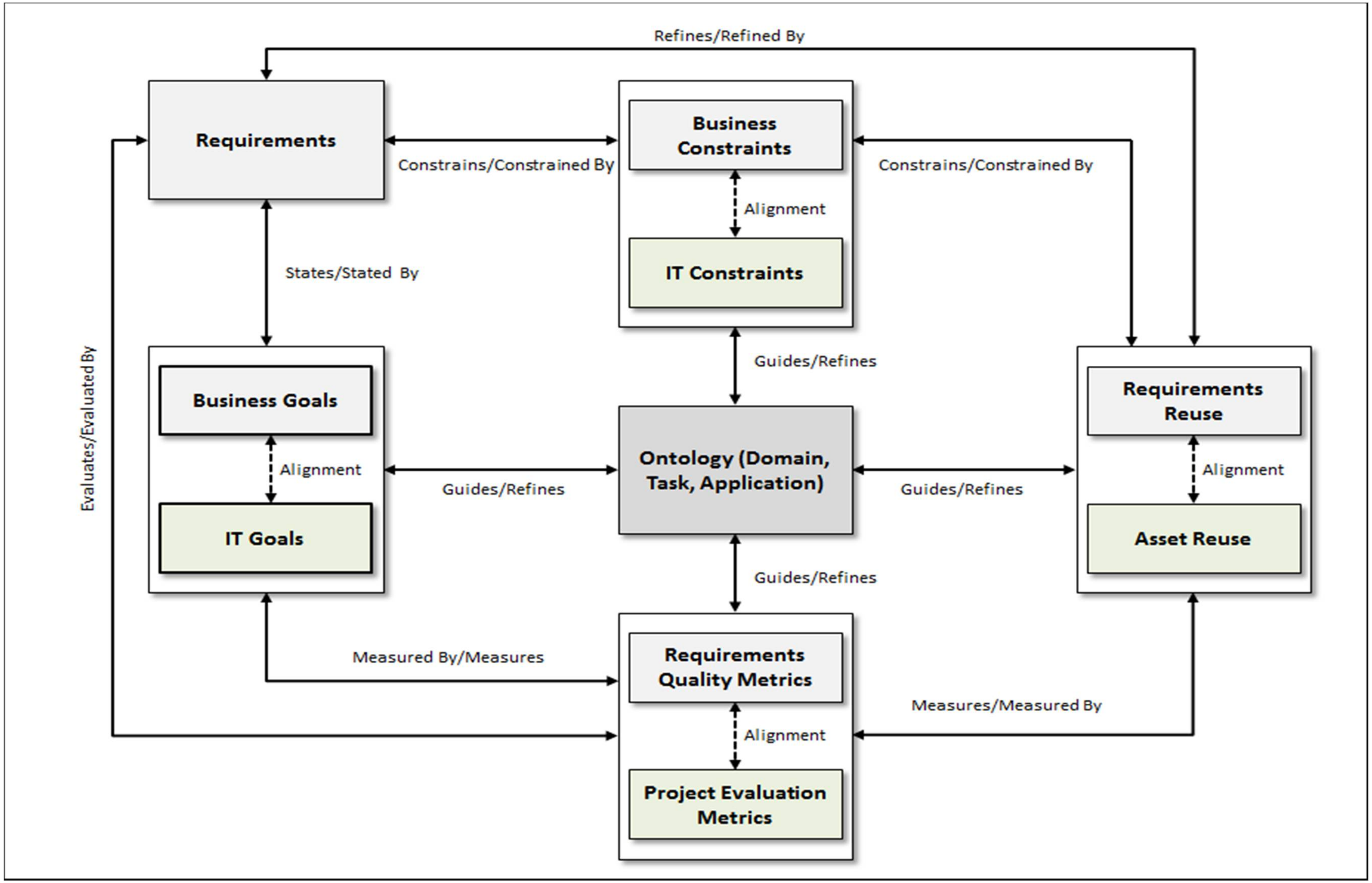

Figure 5.1: REFINTO Framework

Table 5.1: REFINTO Framework Project Stage Mapping to SDLC and PMLC

Stage	Description	SDLC	PMLC
Pre-Project	Activities performed in preparation to project implementation begins	Requirements Engineering	Scoping, Initiating, Planning
Intra-Project	Activities performed during project implementation	Design, Development, Testing, Deployment	Execution, Launching, Implementation, Monitoring, Control
Post-Project	Activities performed after project implementation is completed	Verification, Maintenance	Closeout, Evaluation

The REFINTO framework operates on a *per-project* basis. This implies that the framework treats each project as an instance of a measurable business-IT alignment improvement effort. This is important because the framework operates at the functional level of the enterprise - the tactical and operational level. It is assumed that cumulatively, business-IT alignment attained in individual projects at the tactical and operational levels aggregate to business-IT alignment maturity at that level. This is expressed in equation:

$$\mathbf{B\ (t/o) = (Ba + Bb + \ldots + Bn)/n.} \qquad \mathbf{(1)}$$

Where Ba…Bn are alignment measures of individual projects and B(t/o) is the average business-IT alignment at tactical/operational levels obtained from the aggregation of alignment scores for a portfolio of projects.

5.2.1 REFINTO Framework Workflows

The REFINTO framework workflow can be considered from the perspective of the process prescribed in the framework or the role stakeholders play within the process. These are presented in this subsection.

5.2.1.1 Role-based Workflow

In the role-based workflow depicted in Figure 5.2, the roles and activities collaboratively performed by stakeholders (actors) within the REFINTO-guided process for requirements engineering, management, project execution, and evaluation is described.

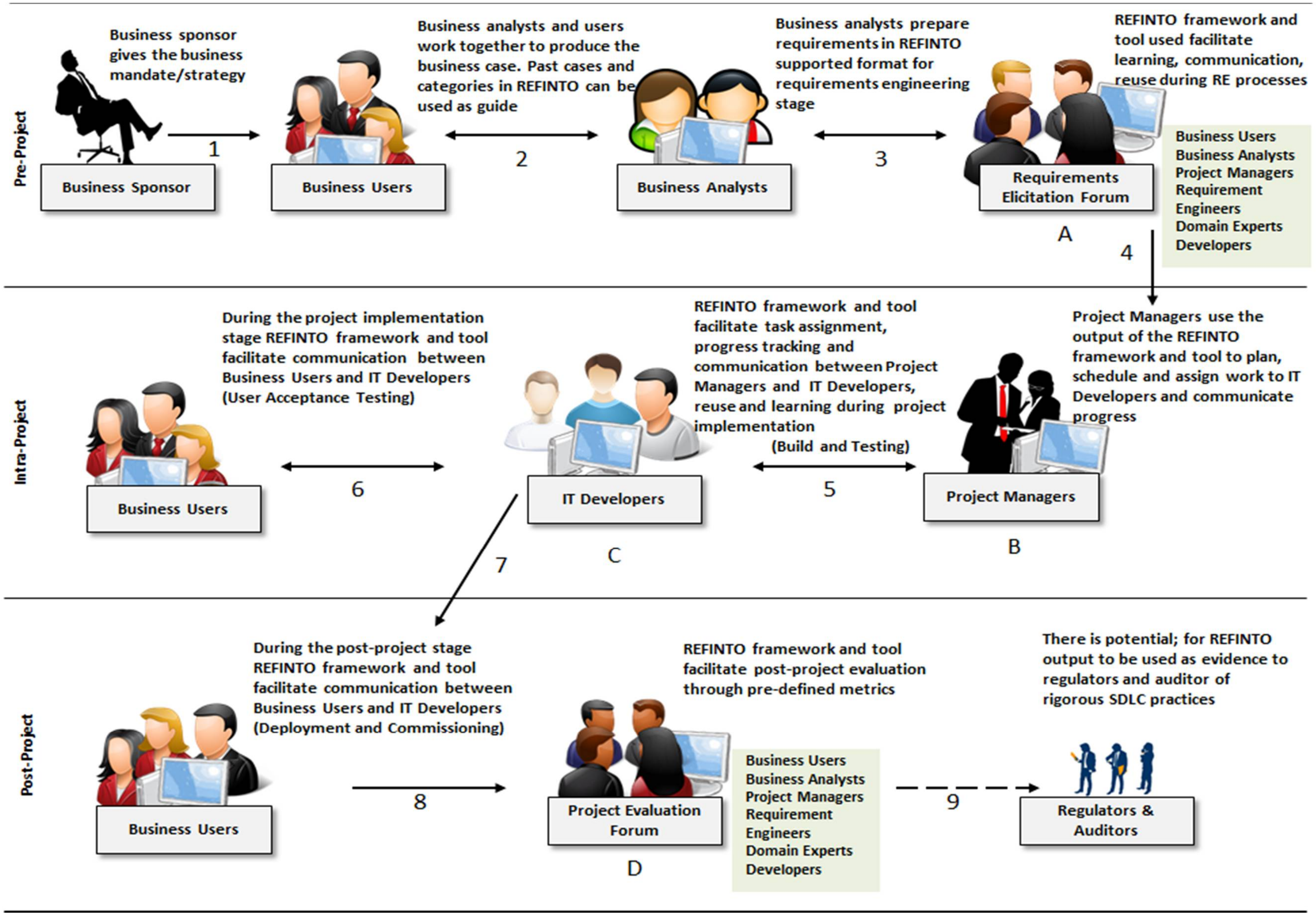

Figure 5.2: REFINTO Role-based Workflow

The importance of tailoring the framework to the tasks performed by these actors is supported by Ciborra (Ciborra, 2000) who argues that whilst there are no set of rules to control alignment, linking multiple actors and resources by balancing complexities and interdependencies can be useful.

The stakeholders defined in the framework include, but are not limited to, business sponsors, business users, business analysts, subject matter experts, requirements engineers, project managers, and IT developers. Some of the actors are passive in the process. This includes business sponsors, regulators, financial managers, and internal/external auditors. Other actors are active to varying degrees.

The framework facilitates collaborative forums namely requirements elicitation forums where business and IT stakeholders work together at the pre-project stage, project manager and developer forums at the intra-project, and the post-project evaluation forum. These forums can also facilitate a *viewpoints* approach to requirements engineering (Sommerville and Sawyer, 1997b). The forums are signposted by letters (A to D) and transitions between the roles in numbers (1 to 9). These forums and transitions are explained here:

A. *Requirements Elicitation Forums*: Business analysts and requirements engineers work with business users (domain experts) to get initial requirements in formats that can be readily used in the REFINTO framework process and support tool. They are responsible for running the requirements elicitation forums. They are active users of the REFINTO framework support tool and the more technically astute ones can typically double as knowledge engineers responsible for ontology and knowledgebase maintenance.

   Project Managers are integral to the requirements elicitation forum from a project management perspective. Domain experts provide advice based on their experience and skills. They provide answers to the questioning process that the framework poses. They can also provide guidance and make decisions on constraints related to requirements and assets. IT Developers are useful actors in this forum. Their insight, experience, and perspective can highlight technical constraints that can influence the realisation of requirements. This forum can be supported by the requirements and artefact management module (RAMM) of the REFINTO tool or an equivalent tool to provide knowledge-based, semi-automated requirement elicitation, reuse, refinement, persistence, and generation of requirements tasks workbook (demonstrated later in this chapter).

B. *Development and Implementation Forums*: Developers are the primary actors at the intra-project stage. They use the framework support tool or equivalent substitutes for matching requirements with existing assets for reuse. This improves productivity, reduces development duration, testing duration, and maximizes return on investment in IT. The forum also facilitates interaction with business users and project managers to clarify ambiguities and to report progress and highlight obstacles. This gives stakeholders visibility of the implementation process and is important to business-IT alignment maturity. This forum can be supported by the requirements and artefact management module (RAMM) of the REFINTO tool, for requirement-artefact matching to facilitate reuse and minimize duplication of development effort.

C. *Project Management Forums*: The project management forums run by project managers is a platform for performing tasks such as estimation, scheduling, monitoring, and progress reporting. It also an opportunity to use the requirements task workbook generated using the REFINTO framework support tool during the elicitation forums to guide the task estimation and scheduling. Equivalent tools like MS Project or other project management tools can also be used. Business, users, business analysts, and IT developers can participate in this forum to ensure accuracy of estimates, schedules, and flag up potential risks.

D. *Project Evaluation Forums*: The project evaluation forum is a platform for collaborative review of the project implementation processes, deliberate on lessons learnt, provide feedback, and document these for future projects. This forum can be supported by the data evaluation and management module (DEMM) of the REFINTO tool presented later in this chapter, by administering role-based surveys to stakeholders depending on the projects participated in and project stage.

Before the forums are held, pre-forum activities are performed by the various actors as highlighted in the description of the transitions. Transitions can be from actor to actor, actor to forum, or forum to forum. The transitions are described here:

1. *Business Sponsors to Business Users*: Business sponsors give directives for or sign off on business cases and needs before business analysts and users initiate processes for the requirements elicitation forum. This ensures that prioritization of business cases has management support and is in line with business strategies.

2. *Business Users to Business Analysts*: Business users collaborate with business analysts, requirements engineer, or project manager from the IT function, to draft the high level business case document that will be used in the requirements elicitation function.

3. *Business Analysts to Requirement Elicitation Forum*: The business analyst, requirements engineer, or project manager prepares the business case document in a format that can be used for semi-automated or manual requirements elicitation, refinement, and persistence.

4. *Requirements Elicitation Forum to Project Management Forum*: The transition involves the transmission of elicited, validated, and prioritized requirements for estimation and scheduling of implementation tasks. The transition can be smoother when project managers and other relevant IT stakeholders are participants in both forums.

5. *Project Management Forum to Development Management Forum:* This transition involves passing prioritized and scheduled tasks to developers for implementation. Developers communicate implementation progress and escalate issues through this transition.

6. *Business Users to Development Management Forum (Implementation):* There is close interaction and collaboration between business users and IT developers during software build and testing phases. The active involvement of business users ensure visibility of implementation progress and facilitate clarification of ambiguities and misunderstanding. This makes it easier to find errors early in the development cycle.

7. *Business Users to Development Management Forum (Deployment):* The involvement, interaction, and collaboration between business users, IT developers, and project managers during post-implementation activities involve reporting bugs and requesting enhancements for future enhancements. In an agile environment, this triggers a repeat of the process to satisfy the requests.

8. *Business Users to Project Evaluation Forum:* Business user involvement in post-project evaluation is important for providing feedback on their perception of the delivered IT artefact and the process of delivery. This facilitates the building of partnerships and bridging of cultural, language, and knowledge gaps between business and IT stakeholders.

9. *Project Evaluation Forum to Auditors and Regulators:* Although regulators and internal/external auditors are not direct actors within the REFINTO framework workflow process, it is increasingly common for them to request proof of structured and rigorous approaches to software development and project management. For example evidence of processes followed for requirements elicitation, testing, and deployment can be requested. This is often a regulatory requirement especially for applications that are expected

to be SOX-complaint[3]. The output of the project evaluation forums can be useful for demonstrating this.

The REFINTO framework role-based workflow already presented and the process-based workflow, presented next, can be executed with the REFINTO tool support or with a third party tool that can provide the functions which include supporting the use of domain ontologies, past projects, to facilitate communication, reuse, learning, and quality control of requirements.

The orthogonal use of the support tool and the framework is discussed later in this chapter. The various scenarios involving the use of the REFINTO framework, ad hoc processes, third party frameworks, with the REFINTO framework support tool, third party tools, or without tools are explored.

5.2.1.2 Process-based Workflow

The initiation point in the process is receipt of requirement(s), which state business goals that corresponding IT goals have to satisfy. At a minimum, these requirements must state input, process, and expected output to satisfy business goals. The process of refining the requirements can then begin in the requirement elicitation forum.

This is an intensively collaborative process that can be performed with or without the framework tool support. The advantage of using the tool is the automation of ontology parsing, question-response process, and logging of responses. The initial requirements are classified based on broad functional grouping, if the requirements are collectively deemed to be appropriate for that grouping.

The process-based workflow of the REFINTO framework is depicted in Figure 5.3.

[3] Sarbanes-Oxley Act of 2002 mandates auditing and validation of IT systems that perform financial reporting functions

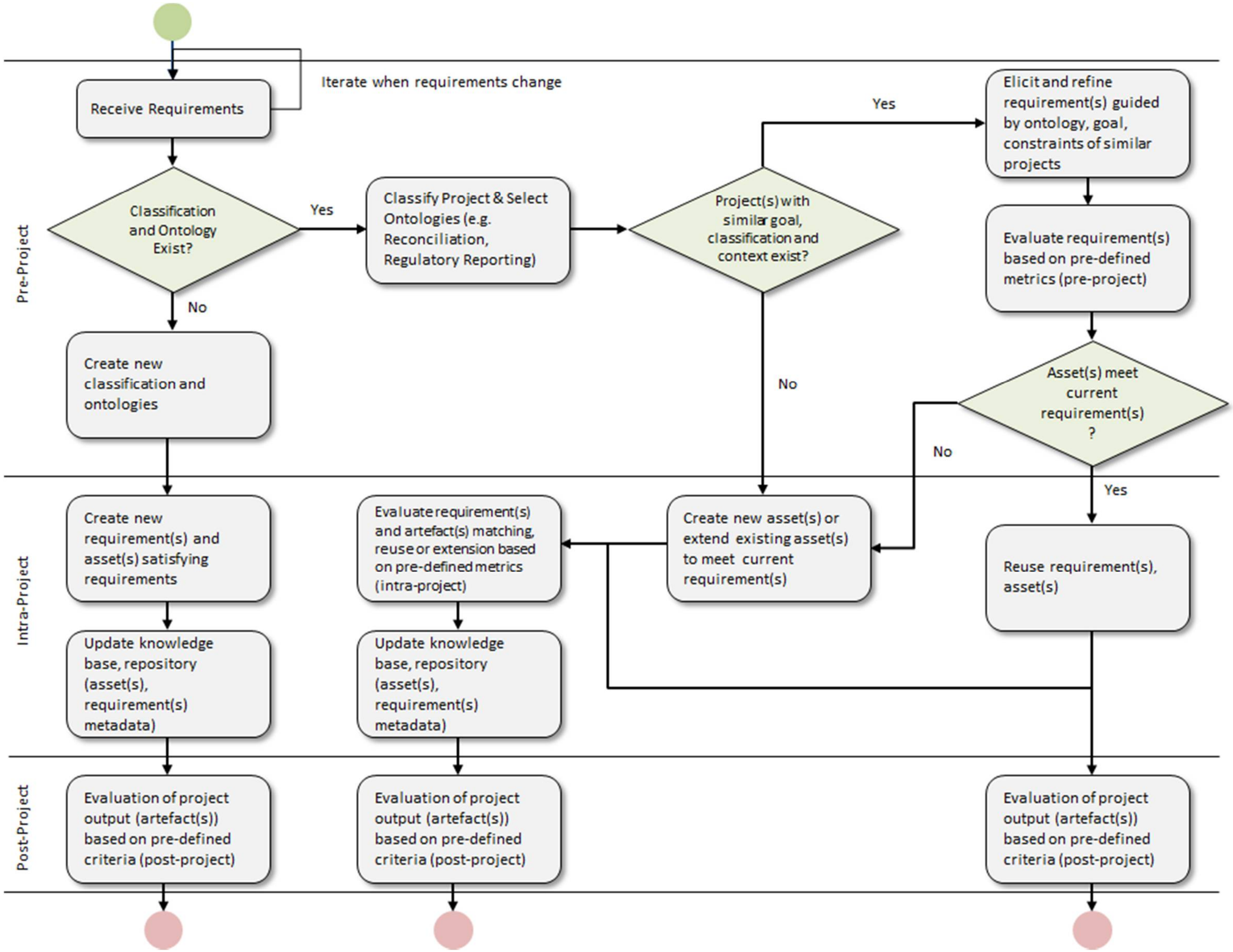

Figure 5.3: REFINTO Process Flow

The framework support tool allows for querying the repository, which contains details of existing assets, automating the process of identifying capabilities for extension or reuse. The evaluation of intra-project alignment indicates how relevant and accurate the process of identifying, selecting, extending, and reusing appropriate assets to meet the current project's requirements was. It also indicates the visibility of the processes, expressed through communication and learning, was to both IT and business stakeholders. This interaction between business and IT stakeholders is also important during development and user acceptance testing (UAT) phases.

On completion of the project, post-project evaluation is performed against pre-defined criteria. Participants in this evaluation are from business and IT. The evaluation at this stage focuses on gauging if IT has delivered the right solution to business within schedule and cost constraints. The REFINTO process-oriented workflow is iterative and therefore addresses the fact that requirements are rarely complete upfront in iterative software development methodologies such as agile and RAD.

5.2.2 Requirement and Asset Matching

The requirement-artefact matching process has five stages namely initiation, matching, identification, refining, analysis and recommendation. An illustration of this process is depicted in Figure 5.4 and is described here:

1. *Initiation*: The initiation stage involves the receipt of the business case or requirements document stating the goal, input, process, output, and known constraints. The requirements are broadly categorized by function based on the domain ontologies. Requirements can fall into multiple classifications.

2. *Matching*: Based on the classification of the requirements, a search for matching historic projects, requirements, artefacts, and constraints is performed. The search for matches is based on the metadata defined for the

historic projects, requirements, artefacts, and constraints. This assumes that a knowledge base and repository of historic projects, requirements, artefacts, and constraints exists. This stage is best performed with some automation because manual search is time consuming. The REFINTO tool provides semi-automated search capabilities. Third party tools that can provide similar functions can be used as substitute. In Figure 5.4, there are two historic cases (case A and B) matching requirement C.

3. *Identification*: This stage involves mining into matching historic projects for requirements, artefacts, constraints that can be used in the requirements refining stage. In the example in Figure 5.4, two historic requirements (requirement A and B) from case B match new requirement C.

4. *Refining*: This stage involves addressing or accommodating constraints that may have been overlooked when new requirements were drafted. This may require making revisions to new requirements. In the example in Figure 5.4, two constraints (constraint A and B) associated to historic requirement A are used to refine new requirement C.

5. *Analysis and Recommendation*. The analysis and recommendation stage typically result in three possible outcomes. Firstly, recommendation for acceptance of new requirements as is or refined and correspondingly creation of artefacts (assets) to satisfy these new requirements. Secondly, existing requirements can be considered similar to new requirements, and the recommendation would be to reuse existing assets to satisfy the new requirements. Lastly, there may be existing requirements that are similar to the new requirements to some extent and therefore have assets that partly satisfy the new requirements that can be modified or reused as is. In the illustration in Figure 5.4, historic requirement A has two features (feature A and B) similar to new requirement C but one feature (feature C) which is not captured in requirement C. Requirement C has a feature (feature D) which is not in historic requirement A. This allows business and IT stakeholders to collaboratively decide on the appropriate option.

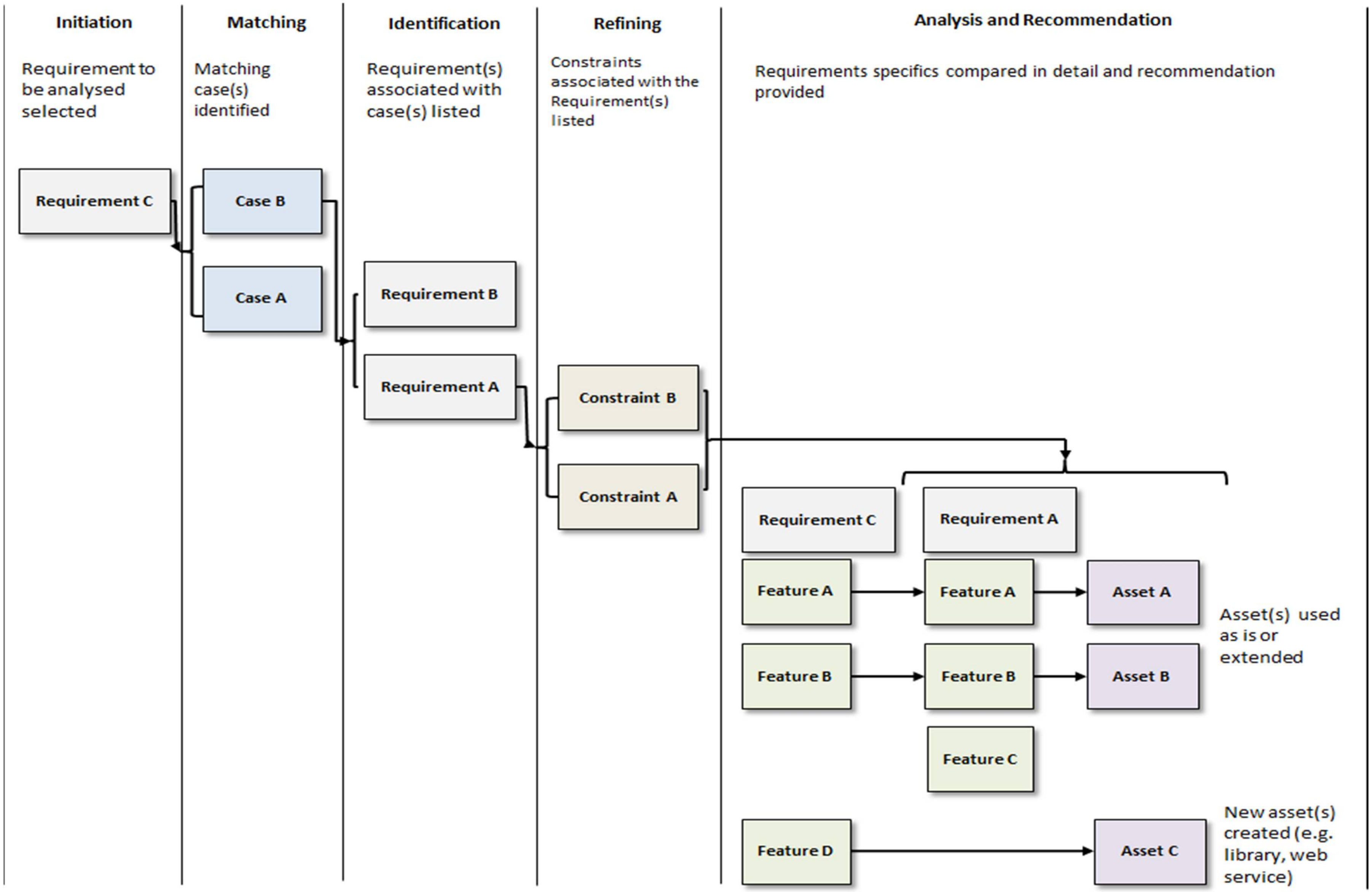

Figure 5.4: REFINTO Case Matching and Analysis

5.2.3 Alignment Measurement and Evaluation

The measurement and evaluation mechanism of the REFINTO framework is based on measures from the strategic alignment maturity model (SAMM) (Luftman, 2000), Balanced Score Card (BSC) (Kaplan, 2008, Kaplan and Norton, 1996), extant literature, and industry practice experience. SAMM is a strategic level (macro) oriented maturity model. At the operational and tactical levels (micro), it is common to use frameworks like COBIT, ITIL and BSC. Many alignment maturity measurement models are based solely on SAMM (Chen, 2010, Khaiata and Zualkernan, 2009, Guiterrez et al., 2006). They leave out valuable factors such as those in BSC, which have potentials to give indications of alignment maturity from the perspective of financial implications and organizational learning.

The combination of SAMM and BSC is a novel approach to alignment maturity evaluation and is used in a business unit-tiered approach to alignment (Ahuja, 2012). The gap in the Ahuja model is that it does not demonstrate how the business unit alignments are measured. The REFINTO framework addresses this weakness and is better suited to the tactical and operation levels. This is because it follows a project-based model where the daily activities in a business unit in the organization can be tangibly measured and evaluated before, during, and after project execution.

The evaluation mechanism results in individual scores for these stages which, are then aggregated to obtain the overall scores for the project. The results of the analysis of projects are then classified according to the levels formulated which are closely aligned to SAMM/ITGI maturity ranking levels. The evaluation is based on criteria selected for the framework from SAMM, BSC, extant literature, and practice experience, considered relevant, appropriate to objectively and tangibly measure alignment maturity, and relate to the eight hypotheses proposed as antecedents to operational and tactical business and IT-alignment. A high level illustration of the project-based and level-tiered alignment measurement and evaluation model is depicted in Figure 5.5.

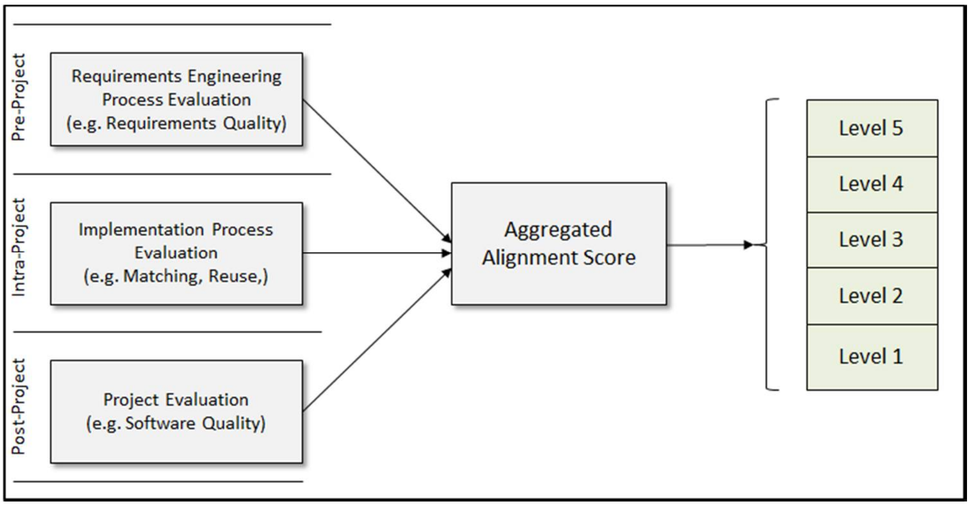

Figure 5.5: REFINTO Alignment Measurement and Evaluation Model

5.2.3.1 Pre-project Stage Alignment Measurement and Evaluation

The focus of the pre-project stage alignment measurement and evaluation is on the requirements quality and the requirements elicitation process. The requirements quality measurement and evaluation metrics include lines of texts, imperatives, continuances, weak phrases, completeness, options, directives, and volatility (Samanta, 2007). Requirements quality follows a 5-point scale described here:

1. *Level 1:* Very low quality requirements, very high volatility, no directives, incompleteness, no continuances, non-optimality of lines of text, mostly weak phrases, too many options leading to ambiguity etc.

2. *Level 2:* Low quality requirements with high volatility, few directives, some incompleteness, few continuances, low optimality of lines of text; weak phrases are common, many options leading to ambiguity etc.

3. *Level 3:* Requirements of average quality, some control procedures in place, directives appear where appropriate, mostly complete/occasional incomplete requirements, and continuances appear where appropriate, there is optimality in lines of text, occasional weak phrases, occasional options with potential to cause ambiguity etc.

4. *Level 4:* High quality requirements, version control is good, less volatility, less incompleteness, no directives; continuances appear where appropriate, optimal lines of text, no weak phrases, where options exist most likely will not cause ambiguity etc.

5. *Level 5:* Very high quality requirements, excellent control of requirements content and version is established, best practices followed in formulating requirement, directives where appropriate, completeness, optimal lines of text, absolutely no weak phrases, no options leading to ambiguity etc., established elicitation process for business and IT.

5.2.3.2 Intra-Project Stage Alignment Measurement and Evaluation

The measurement and evaluation of alignment at the intra-project evaluation stage focuses mainly on the effectiveness of the process of identifying, selecting, extending or reusing existing IT artefacts for current needs. This is based on the argument that significant time and cost savings can be made in project by extending, reusing, or modifying existing requirements and IT artefacts to satisfy new requirements. This saves time and effort that would otherwise be spent on reinventing the wheel.

The REFINTO framework facilitates reuse through identification of historic projects relevant to current needs and reuse of the IT artefacts delivered to meet the matching those requirements. It also facilitates documentation and classification of capabilities and metadata of developed IT artefacts to ease future reuse. It also mandates a disciplined and systematic approach to reuse, modification, and knowledge capture about requirements and IT artefacts. An ad hoc approach to these activities is error prone and the chances of duplication of effort are greater.

Ranking of intra-project alignment maturity measurement follows a 5-point scale. The ranking at this stage is more straightforward than the pre-project stage. For example, if a respondent found the matching and identification for reuse process very useful it would be 5 and if it not useful at all it would be assigned a score of 1.

The mean scores of the sub-processes of identification, relevance, documentation, learning, communication, and resource efficiency are aggregated to obtain the alignment score for the intra- project stage (Umoh et al., 2012).

5.2.3.3 Post-project Stage Alignment Measurement and Evaluation

The focus of the post-project stage alignment measurement and evaluation is on the perception of the stakeholders of the process and outcome of the project against agreed deliverables. The metrics for the post-project stage are based on software estimation and project evaluation metrics such as software size (planned and actual number of units, lines of code), staffing (planned and actual levels over time), complexity of each software unit, progress (planned and actual milestones achieved), problems/change status report containing total number of issues, number closed, number opened in the current re-porting period, age and priority (Chan et al., 2006).

Other metrics used include assessment of planned delivery dates and milestones comparison to actual dates and milestones, build release content (planned and actual number of software units released in each build), and actual resource utilization versus planned resource utilization captured in business case. Discard or rework of IT artefacts to satisfy requirements is evaluated. Also evaluated are learning, resource efficiency (time/financial) made, and effectiveness of communication through the project lifecycle.

The ranking is also on a 5-point scale. For example if a respondent agrees that planned and actual dates for key deliverables were met as scheduled a score of 5 is given and if all key deliverables were not delivered as scheduled, a score of 1 is given. It is expected that alignment scores obtained at the post-project stage correlates with the scores from the pre-project and intra-project stages. This is based on the argument that the quality and process of the requirements elicitation process has an effect on the outcomes of the project (Umoh et al., 2012).

5.2.3.4 REFINTO Project-based Alignment Measurement Model

The project-based measurement and evaluation model is depicted in Figure 5.6. It illustrates how the various metrics (*Criteria a….n*) for a parent metric (Quality for instance) is aggregated to obtain alignment scores for a stage (Pre-project for instance) and how the scores for the three stages (pre-project, intra-project and post-project) of a given project aggregates to the alignment score for that project. Finally it illustrates that the alignment scores for a portfolio of projects can be aggregated to obtain the alignment for that portfolio of projects.

This model is significant in two regards. Firstly, it is used for the data collection and analysis of the framework. Secondly, it indicates that the framework is capable of satisfying the same business unit-tiered model proposed in (Ahuja, 2012). The portfolio of projects depicted in the illustration in Figure 5.6 could be all projects from a given business unit and the alignment scores obtained would indicate the business-IT alignment maturity for that business unit.

1. *Scores for Criteria Items*: Each antecedent (hypotheses) has metrics associated with it. Each of metric is measured by question(s) in the data collection instrument.

2. *Scores for Antecedent (Hypotheses):* The scores for antecedents are the mean score of its criteria. The scores for the antecedents can be calculated vertically (for all three project stages) or horizontally (for a project stage particular stage).

3. *Scores for Project Stage:* The scores for each project stage can be calculated. Scores in earlier stages (pre-project and intra-project stages) are expected to correlate with scores at a later stage (post-project stage).

4. *Project Scores:* The aggregated scores for a project by antecedents and project stages can be calculated

5. *Project Portfolio Scores:* The aggregated scores for projects in the portfolio by antecedents and project stages can be calculated.

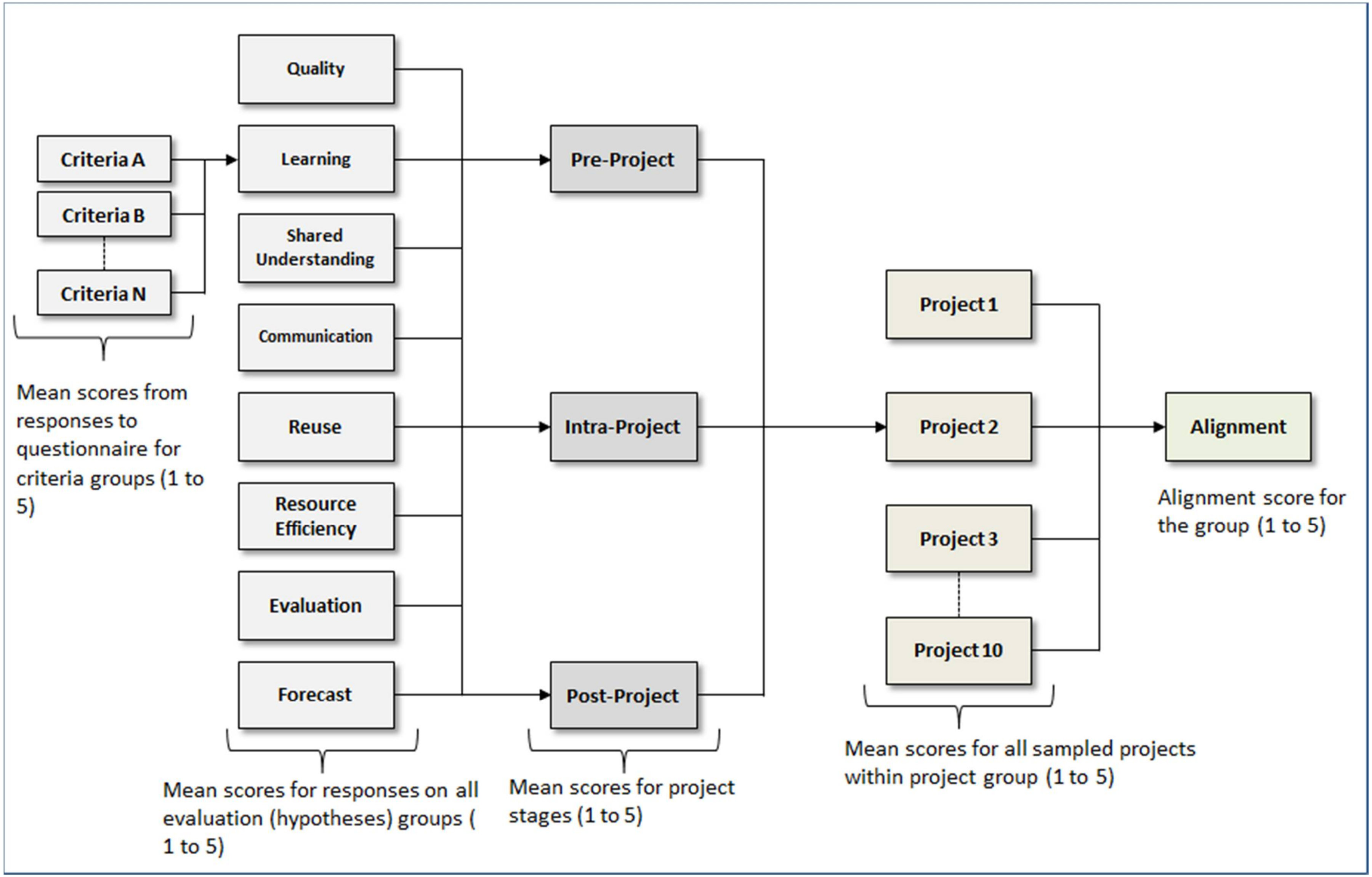

Figure 5.6: REFINTO Project-based Measurement and Evaluation model

5.2.4 Components of REFINTO Framework

The conceptual foundations of the framework such as requirements, reuse, ontology, metrics, and evaluation model are described in this section.

5.2.4.1 Requirements

A requirement can be said to be a realizable goal to be satisfied by agents in a software solution, expressed in terms of monitored and controlled objects. As a component of the REFINTO framework, requirements are first class citizens, an input, and an output of the pre-project stage of the framework. Requirements can be considered as the primary *operands* of the framework. The other components of the framework can be considered as *operators* that transform requirements throughout the software development and project management lifecycles by transitioning them from elicited requirements through to satisfied requirements.

5.2.4.2 Goals

Goal modelling is commonly used in requirements engineering. An example of a goal modelling framework used in requirements engineering is KAOS (Lamsweerde and Letier, 2000, Dardenne et al., 1993). Other goal modelling frameworks such as i* (Yu, 1997) and Tropos (Castro et al., 2002) are considered to be agent-oriented and provide a structure for asking *why* questions in RE. A goal is essentially a prescriptive statement of intent about some system either existing or to be developed. Goals are objectives that the system under consideration should achieve and intended properties that should be provided (Lamsweerde and Letier, 2000, Zave and Jackson, 1997).

Goals are essential in the RE process for achieving completeness, avoiding irrelevant requirements, facilitating requirement explanation to stakeholders, requirement pertinence, traceability, structuring requirements to enhance readability and conflict management (Lamsweerde and Letier, 2000, Lapouchnian, 2005). It has been argued that "*a goal under responsibility of a single agent in the software-to-be becomes a requirement whereas a goal under responsibility of a* *single agent in the environment of the software-to-be becomes an assumption*" (Lamsweerde et al., 1998). Assumptions are not enforced in the envisaged software solution, quite unlike requirements that must be satisfied. Based on these arguments, in the context of the REFINTO framework, requirements can be considered as expressing business goals that IT goals have to align to.

5.2.4.3 Constraints

Constraints can either be business or IT related. Business constraints include regulatory, audit (external or internal) stipulations, policies, thresholds or boundaries that are to be adhered to. These constraints can have impact on meeting business needs. From an IT perspective, there may be issues like insufficient infrastructural capabilities that may put a constraint on satisfying requirements. Constraints can generally have implications for the realisation of functional and non-functional requirements. Awareness of, adherence to, or identifying workaround these constraints are critical to capturing and satisfying these requirements.

In the context of the REFINTO framework, using known constraints from existing satisfied requirements to refine new requirements has potential benefits. For example, identifying these constraints early in the SDLC can help to reduce the waste of build time and associated costs further along in the and be a medium for learning about the domain.

5.2.4.4 Reuse

Reuse in the context of software engineering and requirements engineering specifically has been discussed in chapters two and three. Reuse is an important component of the framework. Valuable business knowledge captured in past requirements can be reused as a template to refine or validate current requirements. Refining new requirements implies adopting aspects of existing requirements that have been satisfied as is or extending current requirements with information from historic requirements that have been satisfied that may have been inadvertently missed or overlooked.

5.2.4.5 Metrics and Evaluation

Definition of metrics for the eight antecedents to operational and tactical business-IT alignment applicable to gauging alignment maturity is a critical component of the REFINTO framework. In chapter two, the antecedents to business-IT alignment from various literatures were reviewed. The limitation of most of these antecedents is that they are difficult to operationalize. This is because they are targeted mainly at strategic alignment.

The metrics defined for the REFINTO framework are selected with empirical measurement and the ease of operationalization in mind. The focus on metrics that allow for business-IT alignment maturity measurement to be made at functional levels of an enterprise through project planning, conceptualization, execution, and governance has advantages. This is because it provides a means to tangibly measure alignment in the daily operations of an organization.

5.2.4.6 Ontology

The application of ontologies in software engineering and requirements engineering specifically has been discussed in the literature review chapters. The domain ontologies defined for the business functions in financial services are used to guide the requirement elicitation, requirement classification based on business function categorization, driving the requirement and asset matching process, and capturing constraints. Requirements engineering is a knowledge-intensive activity involving capturing and representing knowledge from many sources and ontologies are suited for this purpose.

Domain ontologies are therefore first class citizens of the REFINTO framework. Ontologies are useful for representation of requirements models, acquisition structures for domain knowledge, application domain, environment and reusable models that are relevant to requirements (Dobson and Sawyer, 2006, Kof et al., 2010, Sutcliffe and Sawyer, 2013). The domain ontologies developed for the framework are presented in Appendix B.

5.3 REFINTO Framework Hypotheses

The eight hypotheses of the REFINTO framework are considered as antecedents to operational and tactical alignment maturity and the pillars for achieving sustainable alignment in the enterprise as depicted in Figure 5.7. It has been highlighted in literature review chapters that it is increasingly being realised that to attain sustainable business-IT alignment, micro-foundational alignment should be considered. These pillars identified in this section are building blocks to achieve micro-foundational alignment.

The hypotheses form the study model used for validating the framework. The hypotheses were identified through extensive review of extant literature and industry experience gained through participation in projects involving close interaction between with business and IT stakeholders.

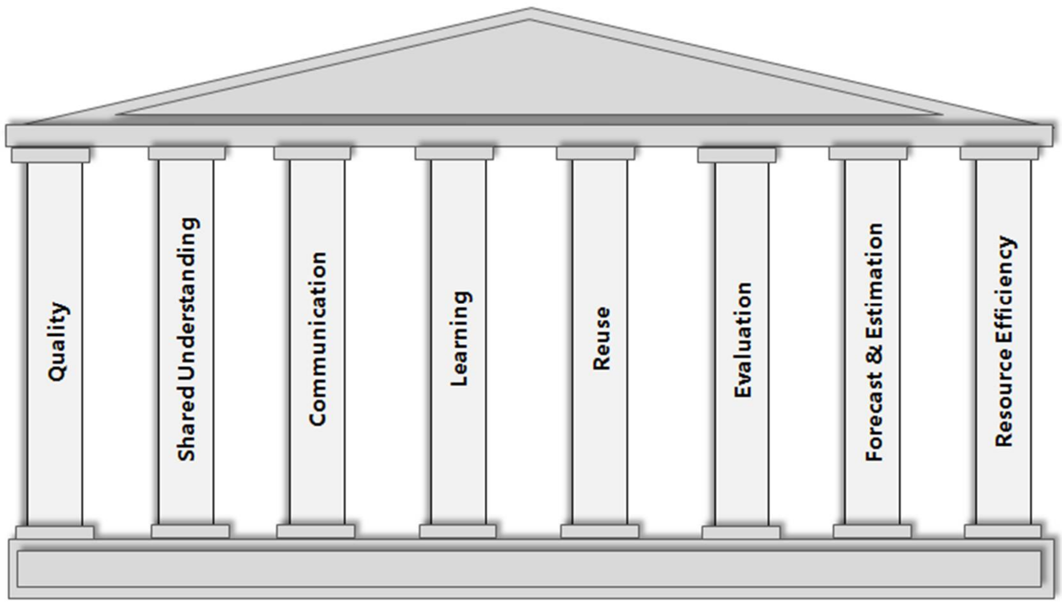

Figure 5.7: Pillars of Sustainable Business-IT Alignment

The study model based on the eight hypotheses is shown in Figure 5.8. This model is validated in the data collection, analysis and interpretation chapters.

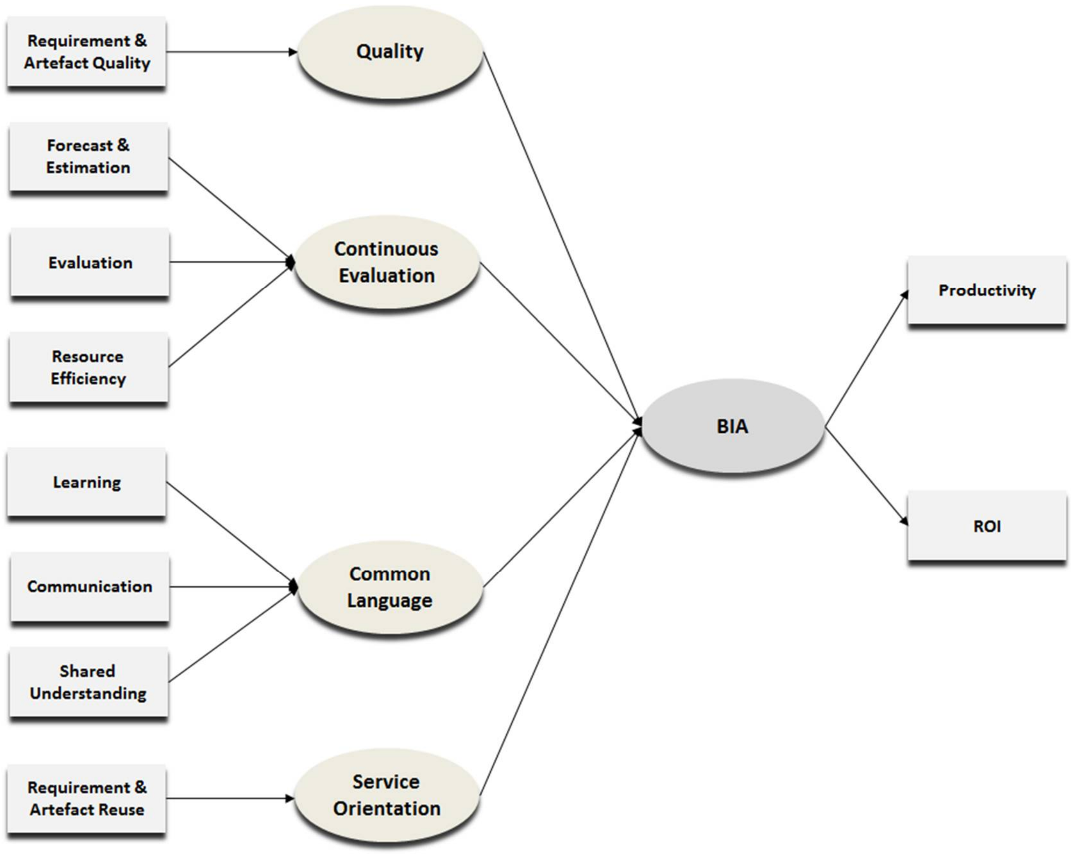

Figure 5.8: REFINTO Study Model

The eight hypotheses can be grouped into four broad categories as shown in Table 5.2.

Table 5.2: REFINTO Antecedent and Groupings

Antecedent	Grouping
H1- Requirement and Artefact Quality	Quality
H2- Shared Understanding	Common Language
H3- Communication	Common Language
H4- Learning	Common Language
H5- Requirement and Artefact Reuse	Service Orientation
H6 - Evaluation	Continuous Monitoring
H7- Forecast and Estimation	Continuous Monitoring
H8 – Resource Efficiency	Continuous Monitoring

5.3.1 Quality

It has been argued that requirements quality has an impact on the quality of the artefact implemented based on the requirements (Knauss and El Boustani, 2008, Radlinski, 2012, Tamai and Kamata, 2009). The impact of requirement volatility, a factor of requirements quality, has on overall software project performance, has been empirically proven (Zowghi and Nurmuliani, 2002, Pfahl and Lebsanft, 2000). The link between requirements quality and business-IT alignment is not readily apparent but is important nonetheless. This link becomes clearer when business requirements and developed IT artefact are examined more closely.

Establishing what business requirements are is a pre-requisite for IT project initiation. The requirements engineering stage (eliciting and managing business needs) of a project is critical to the success of IT projects. This is the stage at which interaction between business and IT needed to ensure clarity of business needs is most intense. It is also the point at which modifications to the business needs have the least impact on cost and delivery time. Therefore, collaboration between business and IT stakeholders to produce high quality requirements and the processes followed to produce those requirements can be critical to the success of the project.

The business purpose for which requirements are produced and the developed IT artefacts which satisfy these requirements can be broadly categorized. New business needs tend to be variations or composite of business needs that may have been addressed in the past. This implies that there are re-occurring patterns. This makes it possible to build a domain ontology that is useful in facilitating requirements elicitation during the prior to project implementation stage or when requirements change. The argument has support in extant literature (Ullah and Lai, 2013, Bleistein et al., 2004, Kujala et al., 2005, Lam et al., 1997).

This forms the basis of the first hypothesis (H1):

Hypothesis 1: *The quality of business requirements has an influence on software product quality and together are positively associated with business-IT alignment at the tactical/individual and operational/department levels.*

5.3.2 Common Language

The importance of a common language between business and IT stakeholders especially in highly specialized industries like financial services cannot be overemphasized. When business and IT speak a "common language" , this can lead to shared domain knowledge and organizational learning (Luftman, 2000, Reich and Benbassat, 2000). This is particularly imperative at operational and tactical levels where frequent interaction between business and IT stakeholders occur. Shared understanding and knowledge of a domain, communication, and learning are essential to engendering a common language. These are explored further and hypotheses developed with respect to operational and tactical business-IT alignment.

5.3.2.1 Shared Understanding and Domain Knowledge

To achieve alignment at the operational and tactical levels, it is important that business and IT stakeholders have a shared understanding of business domain concepts and processes especially in highly technical domains. In the financial services domain, for example, there are concepts and terms that are specific to the domain that may be used in requirements documents. These terms have to be expressed in language IT stakeholders with no prior experience or knowledge of the domain can understand and implement.

Conversely, it is necessary for business stakeholders to have an understanding of the processes and challenges that IT stakeholders have to overcome to deliver on business requirements. This mutual understanding may lead to better relationship and partnership between business and IT stakeholders, more realistic demands from the business, and help bridge the cultural gaps between business and IT (Yayla and Hu, 2009, Chan et al., 2006, Reich and Benbassat, 2000).

This forms the basis of the second hypothesis (H2):

Hypothesis 2: *Shared knowledge and mutual understanding between business and IT is positively associated with business-IT alignment at the tactical and operational levels.*

5.3.2.2 Communication

The link between communication and business-IT alignment has been highlighted (Luftman et al., 1999). Communication is identified as one of the five elements that contribute to short term alignment (Reich and Benbassat, 2000). It is important to have effective communication between business and IT stakeholders at all stages of the PMLC and SDLC. The frequency and relevance of the content of communication between business and IT stakeholders is important. Effective communication facilitates shared understanding, requirements elicitation, and clarification (Teo and Ang, 1999, Huang and Hu, 2007, Yayla and Hu, 2009) and is also essential for proactive and timely resolution of issues, mitigation of risks, and ensuring harmony between business and IT functions.

This is the basis of the third hypothesis (H3):

Hypothesis 3: *Effective, timely, and frequent communication providing visibility of business process, impact of change, issues (delays, constraints etc.) is positively associated with business-IT alignment at the tactical and operational levels.*

5.3.2.3 Learning

Learning can be at an organizational or personal level. Organizational learning through interaction and collaboration between business and IT stakeholders leads to enlightened organizational units. On a group level, learning requires shared interpretation, common language, and mutual understanding (Balhareth et al., 2012). On a personal level, IT stakeholders in specialized domains become more productive as they learn concepts and terms in the domain. Similarly, business stakeholders become better at providing high quality requirements due to improved knowledge of IT processes.

This is the basis of the fourth hypothesis (H4):

Hypothesis 4: *Learning is positively associated with business-IT alignment at the tactical and operational level.*

5.3.3 Service Orientation

Reuse is one of the motivations for service-orientation and other software engineering concepts like object-orientation. It involves using existing IT artefacts such as requirements, constraints, and libraries from historic projects to satisfy current business needs. The artefacts can either be reused as is or modified to meet current requirements leading to time and cost savings. The link between service orientation and business-IT alignment is acknowledged in extant literature (Chen et al., 2008, Beimborn et al., 2009, Selby, 2005).

Whereas reuse of IT artefacts such as libraries is well known, reuse of requirements is less so. Requirements reuse can be beneficial to software development productivity and quality (Lam et al., 1997). It can also be useful as a reference in discussions between business and IT stakeholders for clarifying, refining, and transitioning from what has been requested in the past to what is required now. Reuse (IT artefact and requirements) is therefore considered to be an antecedent to business-IT alignment and is the basis of the fifth hypothesis (H5):

Hypothesis 5: *Reuse of asset, knowledge, and requirements lead to time and financial savings, learning and skills. It is positively associated with business-IT alignment at the tactical and operational levels.*

5.3.4 Continuous Monitoring and Evaluation

Business-IT alignment is a continuous and dynamic process (Gregor et al., 2007). This implies that a mechanism to continuously assess business-IT alignment is necessary (Luftman, 2003, Papp, 1999). The REFINTO project-based alignment measurement and evaluation model presented earlier in this chapter is an objective means of gauging operational and tactical alignment. Evaluating, forecasting, and estimating time/financial savings can influence business-IT alignment maturity at operational and tactical levels. These are discussed briefly and hypotheses derived are presented.

5.3.4.1 Evaluation

Evaluation is one of the three key concepts identified as being relevant to improving business-IT alignment maturity (Aversano et al., 2012). To improve business-IT alignment it is essential that metrics and procedures to continuously measure and evaluate alignment are established. From the perspective of the REFINTO framework, evaluation serves as a medium for business and IT stakeholders to review lessons learnt at the different stages of the software development and project management lifecycles. This forms a feedback mechanism for improving processes thereby improving the chances of success in future system development projects. This is the basis of the sixth hypothesis (H6):

Hypothesis 6: *Continuous evaluation and knowledge gained and codified is positively associated with business-IT alignment at the tactical and operational levels.*

5.3.4.2 Forecast and Estimation

Requirements pattern, requirements reuse, and IT artefacts reuse can be leveraged through a knowledge-based approach to requirements elicitation preferably supported by tool automation and repository of past projects. This can aid the production of high quality requirements and provision of more accurate estimates for systems development projects. It can also make it much easier to more precisely predict project delivery dates, potential pitfalls, and outcomes of current projects. This is referred to as leveraging historic implementation success (Reich and Benbassat, 2000) for business-IT alignment maturity. This can lead to improved alignment between both functions at the tactical and operational levels.

This is the basis of the seventh hypothesis (H7):

Hypothesis 7: *Accurate forecasting and estimation leads to confidence and mutual trust between business and IT functions and is positively associated with business-IT alignment at the tactical and operational levels.*

5.3.4.3 Resource Efficiency

It has been argued that alignment is critical to organizational productivity (Papp, 1999). Resource efficiency in the form of cost and time savings can has been identified as an enabler of business-IT alignment (Silvius, 2007).

The impact of savings as a factor in IT alignment has been empirically tested at strategic levels (Cragg et al., 2002). Resource efficiency as an enabler of business-IT alignment does not only apply to strategic levels. It can also apply to the tactical (individual) and operational (departmental) levels of alignment.

This is the basis of the eighth hypothesis (H8):

Hypothesis 8: *Resource efficiency (in the form of cost and time savings) is has an influence on business-IT alignment at the tactical and operational level*s.

5.4 Design of the REFINTO Framework Support Tool

The objective of the REFINTO framework support tool (RFST) is to provide semi-automated support for the use of components of the framework and the processes defined for practical realization of business IT alignment. This includes the *elicit-refine-persist* and *requirement-ontology-artefact* activities. The overview and design rationale of REFINTO framework and support tool is discussed in this section. The two main components of the tool and its shared infrastructure are presented. An overview of the architecture and innovative features that drive business-IT alignment enablers are highlighted.

5.4.1 REFINTO Framework Support Tool Architecture

The REFINTO support tool follows a multi-tiered architecture as depicted in Figure 5.9.

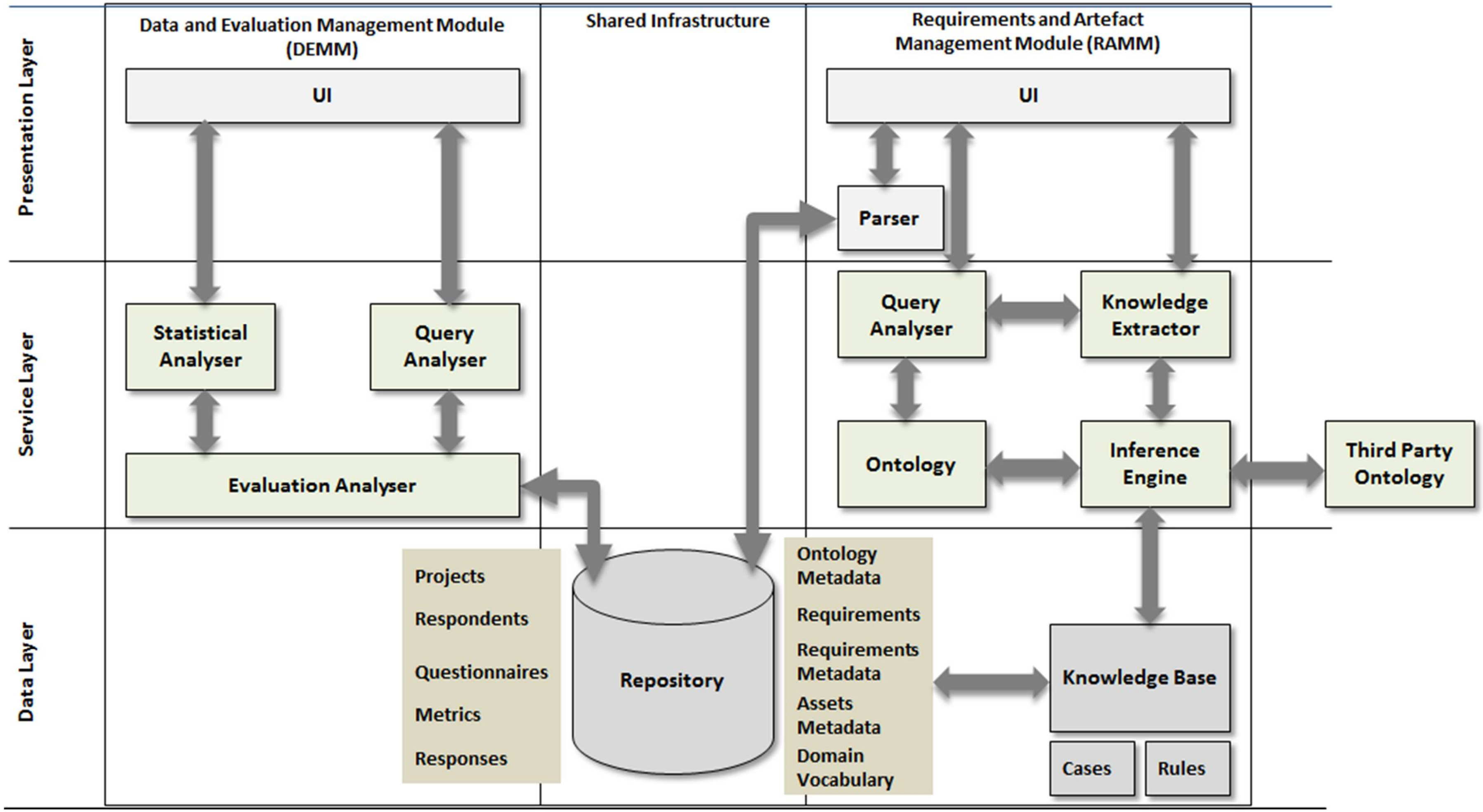

Figure 5.9: REFINTO Tool Architecture

The two modules are the Requirements and Artefact Management Module (RAMM) and the Data and Evaluation Management Module (DEMM). The data layer components (data tables, stored procedures, and libraries for data manipulation and presentation) are shared by both components. The design rationale for separating the modules and developing them in different languages is informed by the need to utilize the strengths and capabilities of both languages for the specialized functions of each module.

5.4.1.1 Requirements and Artefact Management Module

The main purpose of the RAMM module is to provide an interface that both business and IT stakeholders can collaboratively use to elicit, refine and store (persist) requirements to the repository. The graphic user interface (GUI) and control program of RAMM is built on Java/JavaFx following the Model-View-Controller (MVC) pattern depicted in Figure 5.10.

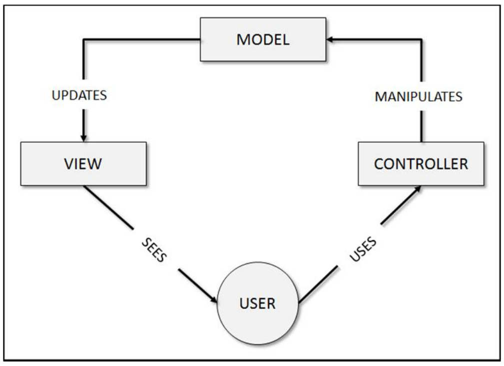

Figure 5.10: MVC Pattern

The business logic (service) layer is made up the inference engine, knowledge extractor, query analyser and the domain ontologies. The control program includes the Requirements Parser which is based on Apache Tika[4] and Apache Lucene[5] for document indexing and content extraction. At the core of this layer are application programming interfaces (API) like Java Expert System Shell (JESS)[6], Pellet

[4] Apache Tika is available at: https://tika.apache.org
[5] Apache Lucence is available at: http://lucene.apache.org/core/
[6] JESS is available at http://herzberg.ca.sandia.gov

Reasoner[7], Fact++, and OWLAPI[8] used in the knowledge engine. This allows for REFINTO domain ontologies to be used in the requirement elicitation process. This semi-automated setup facilitates discovery of assets, reasoning about alignment conflicts and inconsistencies across requirements. The RAMM module user interface is depicted in Figure 5.11. The implementation of the module is discussed later in this chapter.

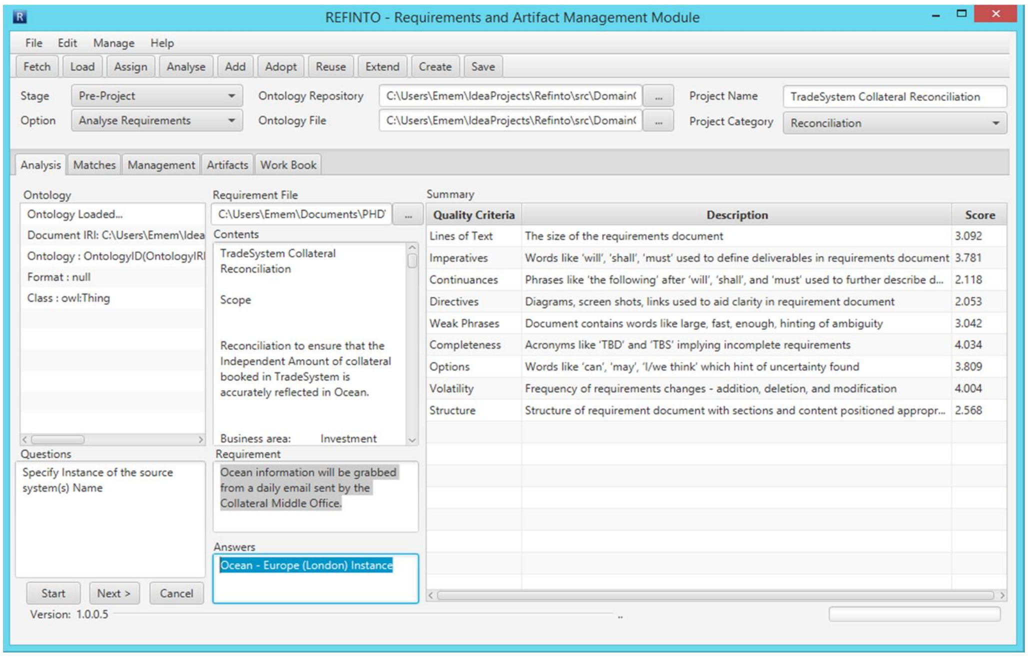

Figure 5.11: Requirements and Artefact Management Module Interface

5.4.1.2 Data and Evaluation Management Module

The primary objective of the DEMM module is to allow users evaluate alignment maturity on a per project basis based on the metrics related to the eight antecedents to operational and tactical alignment. The participation of users in evaluating projects is based on their roles, the projects they participated in, and the stage of the projects. The module also features a statistical analyzer that facilitates calculation and comparison of alignment scores for projects and project portfolios.

[7] Pellet Reasoner is available at : http://clarkparsia.com/pellet
[8] OWLAPI is available at: http://owlapi.sourceforge.net

DEMM is developed in C#/WPF following the Model-View-ViewModel (MVVM) pattern which facilitates the separation of the development of the user interface (view) in markup language (XAML), from the business logic (model). The view model is responsible for exposing the data objects from the model which are easier to manage and consume. This enhances testability and allows for parallel development of the user interface design, model, and view model. The interaction between the view and the view model is through the *DataContext* property of the view which is set to the view model. The view gets updates from the view model facilitated by data binding. The MVVM is depicted in Figure 5.12.

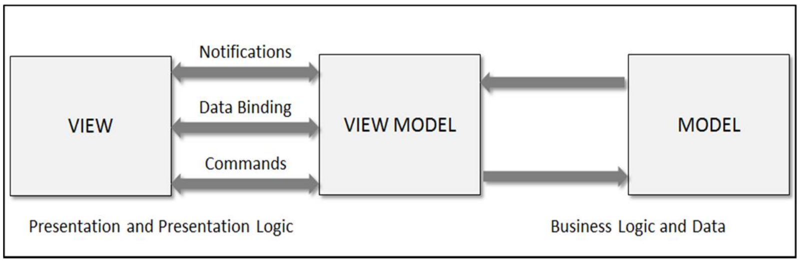

Figure 5.12: Data and Evaluation Management Module Interface

The RAMM module user interface is depicted in Figure 5.13.

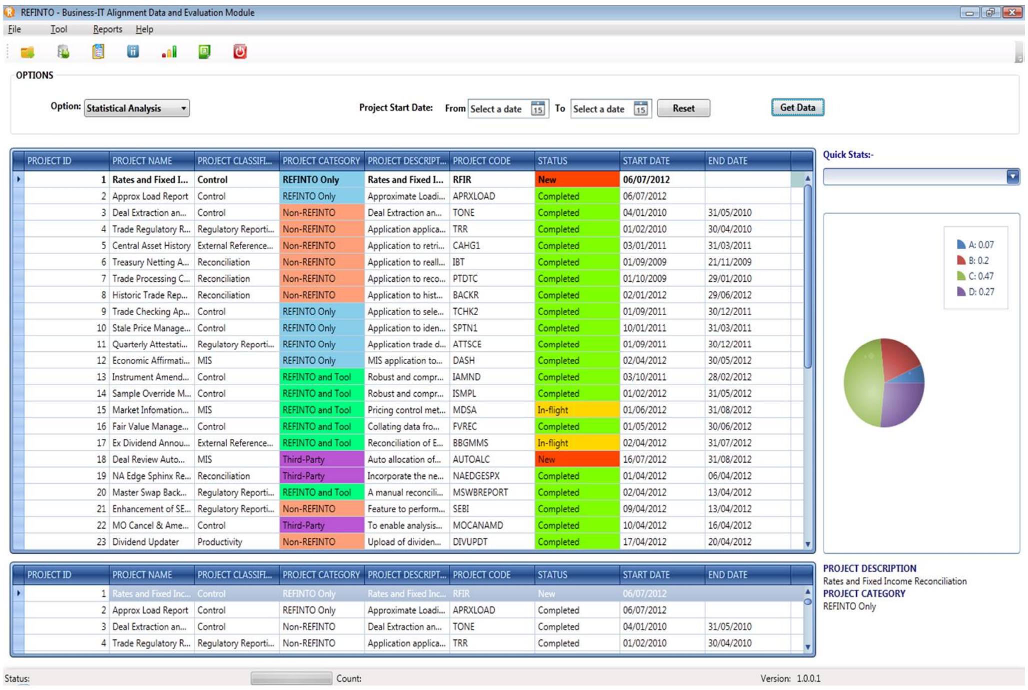

Figure 5.13: Data and Evaluation Management Module Interface

5.4.1.3 Shared Components

The rationale for having a shared data layer is informed by the overlap of data needs between the two modules and the reuse opportunity presented by sharing data captured in both modules. There are various benefits that this architecture offers. It allows for better cross utilization of data, data consistency across the two modules, real time updates, and eliminates duplication of effort through reuse of stored procedures and queries.

5.4.2 REFINTO Domain Ontology

Ontologies are developed for five main reasons namely sharing of common understanding between people and agents, enabling reuse of domain knowledge, making domain assumptions explicit, separating domain knowledge from operational knowledge and analysing domain knowledge (Noy and McGuinness, 2005). The REFINTO domain ontologies were designed with these goals in mind. The ontologies are intended to serve the primary purpose of guiding rigorous and structured requirements elicitation processes towards capturing high quality requirements and facilitating optimal and sustainable business-IT alignment through project execution to satisfy the requirements.

Some the REFINTO domain ontologies form the building block for other complex domain ontologies and are therefore referred to as *common ontologies* in this thesis while those for specific functions are *specialized ontologies*. The domain ontologies developed as part of the REFINTO framework have been subject to revision and refinement throughout the design, implementation, review and application of the framework.

An extract depicting the consolidated REFINTO domain ontology is shown in Figure 5.14. The full set of common and specific domain ontologies are provided in Appendix B. There are also third party finance domain ontologies that are available that can be used where appropriate such as the FIBO ontology (OMG, 2014) also presented in Appendix B.

Figure 5.14: Extract from REFINTO Domain Ontology

5.5 Implementation of the REFINTO Framework Support Tool

The implementation of the REFINTO framework support tool (RFST) realises the n-tier architecture depicted in Figure 5.9. The modules are implemented in three layers namely the presentation layer (user interfaces), business logic (service) layer and data layer. The data layer makes up the shared components used by the two modules of RFST, RAMM and DEMM.

The presentation layer of RAMM is developed in JavaFx following the MVC pattern while the business logic layer is developed in Java for the control flow. The knowledge extraction and inference functions that use the domain ontologies are implemented using Java/JESS and OWLAPI. The presentation layer of DEMM is developed in Microsoft .NET WPF. The business logic is implemented in C# and the statistical analysis uses FoundaStat API[9]. The MVVM pattern followed for the implementation of DEMM handles the interaction between the loosely coupled presentation and business logic layers. Detailed discussion and code snippets for the implementation of RAMM and DEMM are in Appendix A

5.6 Demonstration of the REFINTO Framework Support Tool

In this section, the demonstration of the RAMM and DEMM modules of the REFINTO framework support tool is presented. This is based on the demonstration of the modules made at the demo sessions of the 2014 IEEE EDOC conference (Umoh and Sampaio, 2014). The demonstration is based on the three project stages defined for the REFINTO framework workflows (pre-project, intra-project, and post-project).

The demonstrations show how the two modules are used for requirement elicitation and refinement following an ontology-based reuse mechanism. This mechanism helps to make constraints and requirements linking from historic requirements explicit. It also demonstrates how the tool helps with some aspects of requirement management, prioritization of tasks, status tracking, and generation of tasks workbook for project managers.

[9] FoundaStat is available at: http://www.foundasoft.com/

5.6.1 Demonstration of Pre-Project Stage

At the pre-project stage, the key activities are loading requirements, classifying the project based on the ontology-based classifications, loading the appropriate domain ontology, collaboratively working through the questions that are presented, and refining the requirements. The requirement loading process involves loading the requirements either by reading in the requirements document or typing in the requirements as shown in Figure 5.15. In cases where no domain ontology exists, appropriate ontologies can be created or sourced externally.

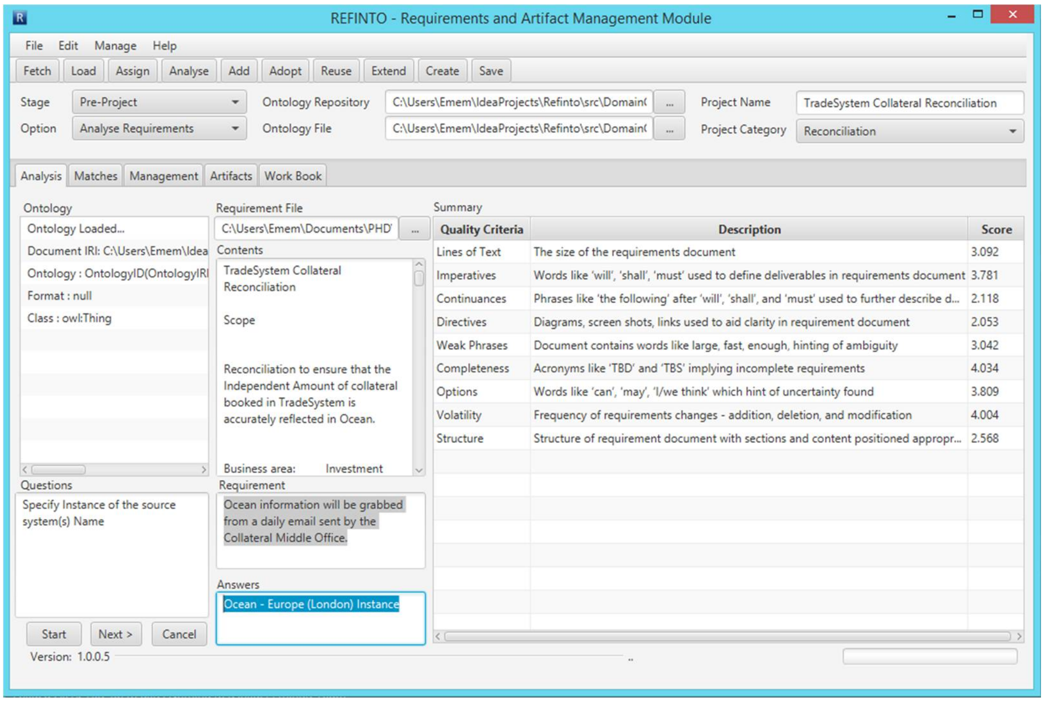

Figure 5.15: Review Matches from Past Projects in RAMM

The next step is to run an analysis on the requirements. This gives an indication of the quality of the requirements based on criteria such as completeness, correctness, ambiguity, continuances, and volatility, etc. The framework and tool allow for the refinement of the requirements based on its reasoning on the domain ontology and matching requirements and constraints from historic cases. Questions relevant to the classification are shown and answers to the requirements elicitation questions recorded.

The next step involves “refinement” and “persistence” of the requirements. Refinement of requirements refers to using domain ontologies and historic requirements to address requirements quality issues like incompleteness, inconsistency, ambiguity, and capturing constraints that may have been overlooked. Persistence of requirements refers to classifying, adding metadata, and storing the requirements to a repository to aid traceability and reuse. The REFINTO framework support tool facilitates the refinement and persistence of requirements as demonstrated in this section, by using domain ontology to guide elicitation, presenting historic requirements and constraints of similar context, and adding unique identification and classification metadata to refined requirements, and saving requirements to the repository (Umoh and Sampaio, 2014).

This *elicit-refine-persist* process is achieved by using the competency questions, ontology walking, cases and rules. This contributes to identifying and addressing gaps in the requirements, violation of business rules, conflicts in the requirements, and omission of essential details for the classification of the project for which the requirements are intended for before the project execution begins. For example, as shown Figure 5.15, the instance (location) of the Ocean question, which is not covered in the requirement, is prompted for.

The elicit-refine-persist process leverages the built-in lexicon of terms in the financial services domain. This is useful in instances where there is a business specific term which is not immediately clear to IT stakeholders. This is a realization of the objective of the framework to bridge knowledge and language gaps between business and IT stakeholders.

The requirement refinement process depends on historic cases codified in the knowledge base. This is done by finding matching cases based on current requirements. The RAMM module allows for collaborative review of these matches displayed on the user interface. Stakeholders can decide to adopt the matched requirement as is, modify the original requirement based on new insights such as business constraints that they were unaware of, or retain the original requirement. A demonstration of this feature is shown in Figure 5.16. In this case, there is a historic requirement that is a match with the current requirement.

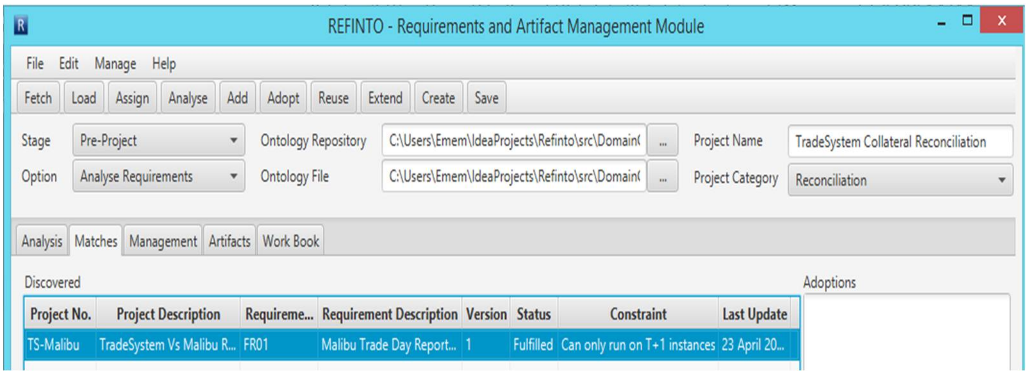

Figure 5.16: Review Matches from Past Projects in RAMM

As can be seen in the screen shot, the historic requirement indicates that there was a constraint stating that the reconciliation could only be run on non-real-time servers (Systems ontology in Appendix B) with data (Data Extraction ontology in Appendix B) that is a day late (Refresh Frequency ontology in Appendix B). The same constraint also applies to the current requirement but was missed. Therefore, the constraint from can be 'adopted' for the current requirement or ignored. In this case the constraint is fundamental and IT stakeholders would not be able to meet the requirement from the business without an express approval from senior management for an exception.

The REFINTO framework support tool automates the persistence of responses, refinement, adoption, or rejection of requirements to the repository. This is critical both for building up cases in the knowledge base for future use, version management, and audit trail. These features are desirable in requirements management tools. RAMM allows the user to either accept or reject a refined requirement. If accepted the requirement is assigned a requirement number. A screen shot showing an accepted is shown in Figure 5.17.

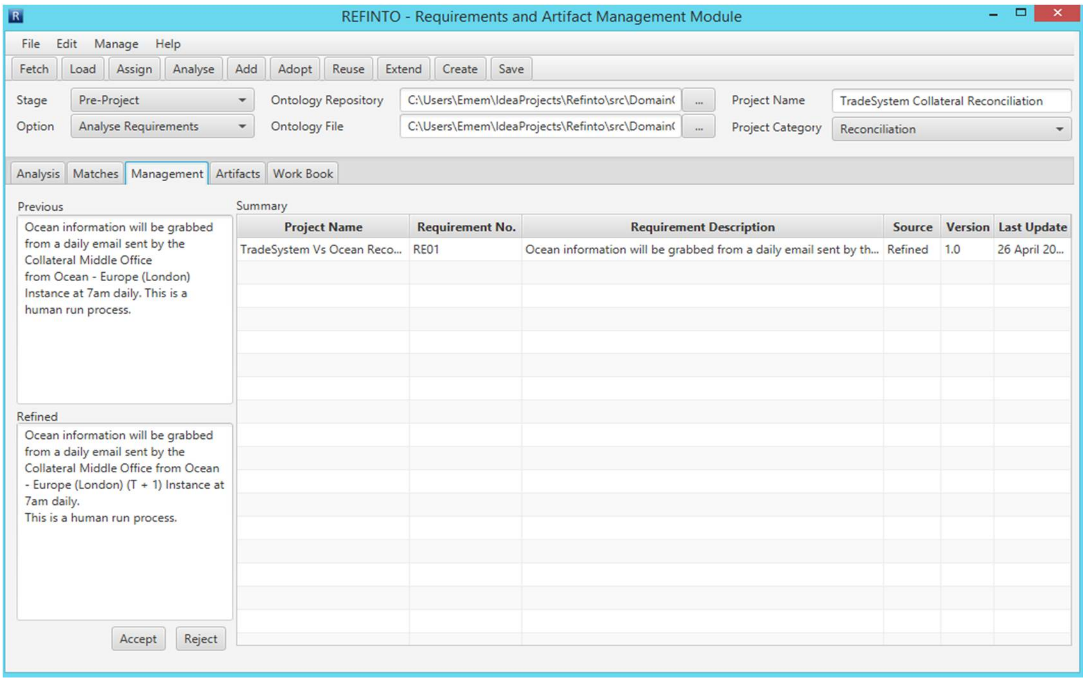

Figure 5.17: Requirements Management in RAMM

Other requirements engineering activities like prioritization and negotiation can then be done and the repository updated with the outcome. At this point the pre-project stage of the process is completed and the evaluation for this stage can be performed by interested parties using DEMM. The intra-project process can then begin with interested parties using RAMM. The elicited requirements are the input of the intra-project stage discussed next.

5.6.2 Demonstration of the Intra-Project Stage

The main activity performed with the REFINTO framework tool at the intra-project stage is of the SDLC is the identification and review of artefact matches. This has two advantages. Firstly, it facilitates reuse of existing IT artefacts developed to satisfy matching historic requirements, either as is or by extending the functions of the assets. This shortens the development duration, eliminates duplication of effort, and facilitates efficiency and productivity improvements. Secondly, it makes it easier for IT stakeholders to provide accurate estimates for tasks to realise the requirements based on estimates provided for similar requirements. The framework support tool allows for the generation of workbook, which contains these estimates.

This facilitates communication with business stakeholders, a key criterion for attaining and sustaining business-IT alignment. It is also a tool that project managers can use for governance functions during the implementation stage.

At the intra-project stage, the framework support tool facilitates review of artefacts that can be used to actualize the refined and accepted requirements. The suggested artefacts most closely matched to the current requirements context as shown in Figure 5.18. Some artefacts are common across project categories. Data retrieval and persistence libraries for example, are common to all project types whereas pricing libraries are mainly specific to pricing functions. The metadata for the specialized artefacts contain the function tag, which is used to link the artefact with the domain ontology for that function.

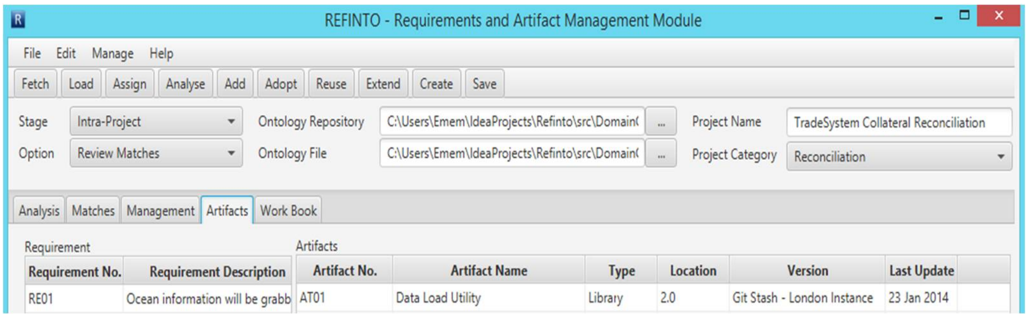

Figure 5.18: Artefact Matches Review in RAMM

The actions that can be taken to satisfy the actualization of the requirements elicited and prioritized in the pre-project stage of the SDLC can be achieved in three ways. These are using existing artefacts for realization of requirements as is, modification of existing artefacts, or creating new artefact. The actions selected for a given requirement and the estimate where necessary is captured in the workbook generated in RAMM. This is shown in Figure 5.19.

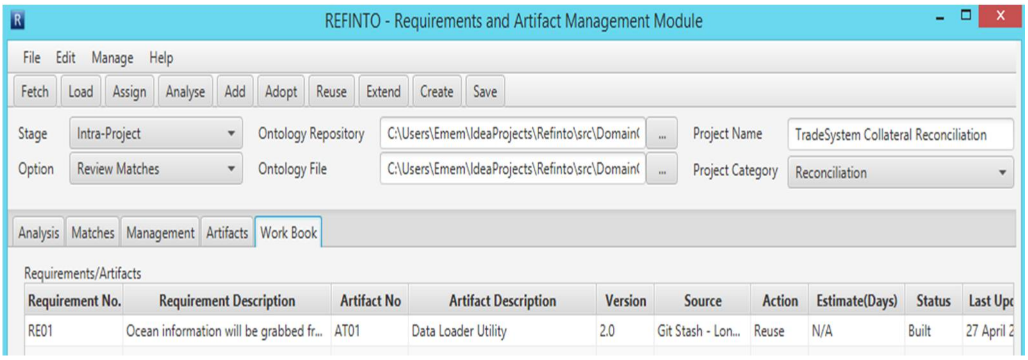

Figure 5.19: Requirements Management in RAMM

As can be seen in the screen shot, an existing artefact has been identified for the requirement. In this case, the decision made is to reuse the artefact as is without any work necessary to make it fit the need of the new requirements. RAMM also captures details of the artefact such as version, location, status, and estimate (in days) for the work to be done, if necessary, to satisfy the requirement is captured as shown in the screen shot.

The estimates for extending or creating artefact are also entered. This level of detail gives stakeholders visibility of the effort and tasks necessary to satisfy requirements. This is an important part of communication criteria for attaining and sustaining business-IT alignment.

5.6.3 Demonstration of the Post-Project Stage

The objective of the post-project stage activities performed with the REFINTO framework and support tool is mainly the evaluation of project, processes, and outcomes. It is a holistic process, which takes into consideration the actual estimates against quoted estimates and the perception of the entire project delivery process. Project evaluation is not strictly a post-project activity. Evaluations are also performed at the end of the pre-project and intra-project stages. A fuller picture is however only available at the end of all three project stages.

The evaluation and ranking functions are performed using DEMM. The evaluation is role-based and respondents can only evaluate projects they participated in.

Participants are given anonymous IDs which are used to relate the projects the participated in, the stage of the project being evaluated, and their responses to the questions presented. This is shown in Figure 5.20.

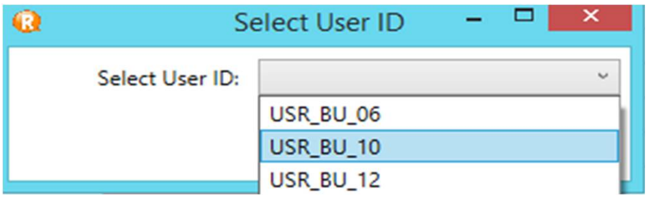

Figure 5.20: Role Selection in DEMM

A screen shot of questions provided to the respondent with user ID ‘USR_BU_25’, a business analyst that participated in the ‘Historic Trade Reporting’ project is shown in Figure 5.21. The user is evaluating the Pre-project stage of the project and the question presented is a requirements quality evaluation question.

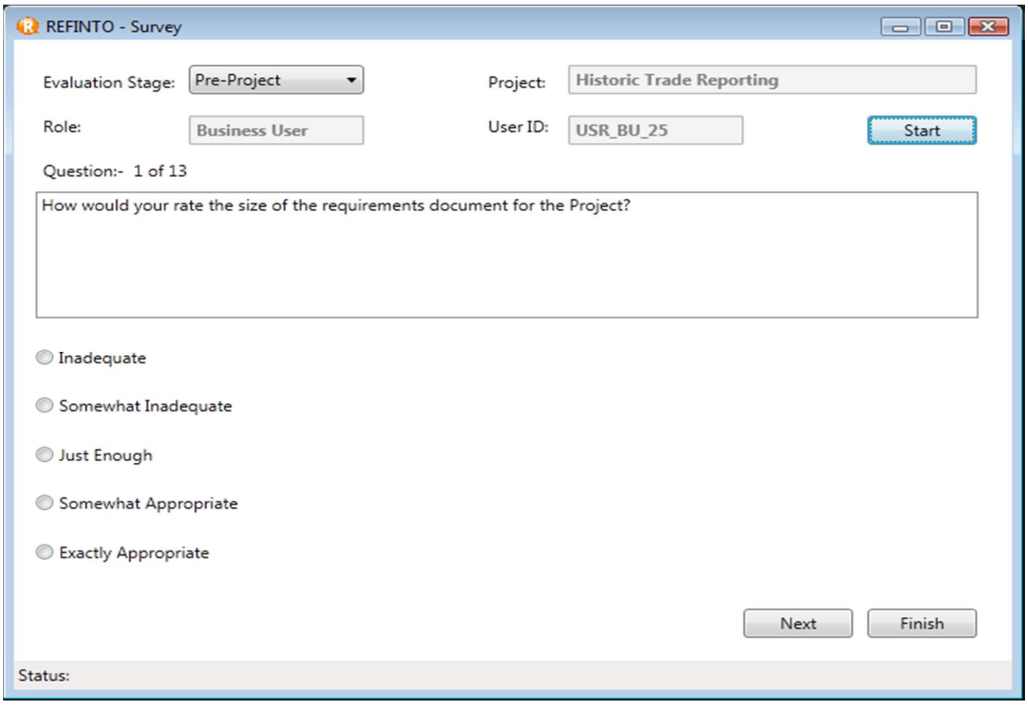

Figure 5.21: Role and Project-based evaluation of projects in DEMM

Participants rank each stage of the project by right clicking the project of interest on the DEMM main window *DataGrid*, depicted in Figure 5.22, and choose to evaluate individual projects or a group of projects by project stage or a combination of the three stages.

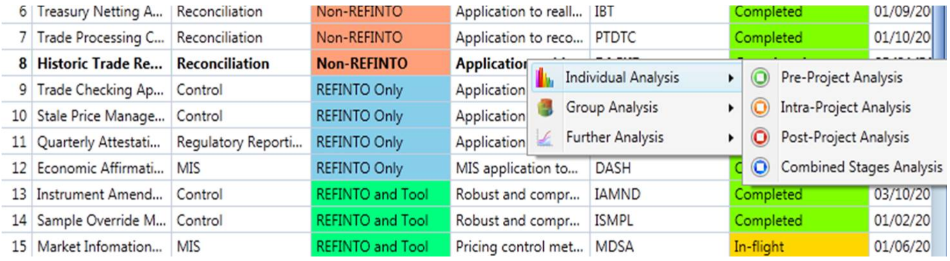

Figure 5.22: Selecting Analysis option in DEMM

The result of the analysis in DEMM correlates directly to the alignment scores for the project. The ranking of project using the statistical analyser in DEMM which implements the project-based measurement and evaluation model discussed earlier in this chapter. A screen shot for an individual project is shown in Figure 5.23 with an arithmetic mean of 3.3187 and would be normalized to alignment score of 3 against the 5 scale REFINTO framework alignment score model.

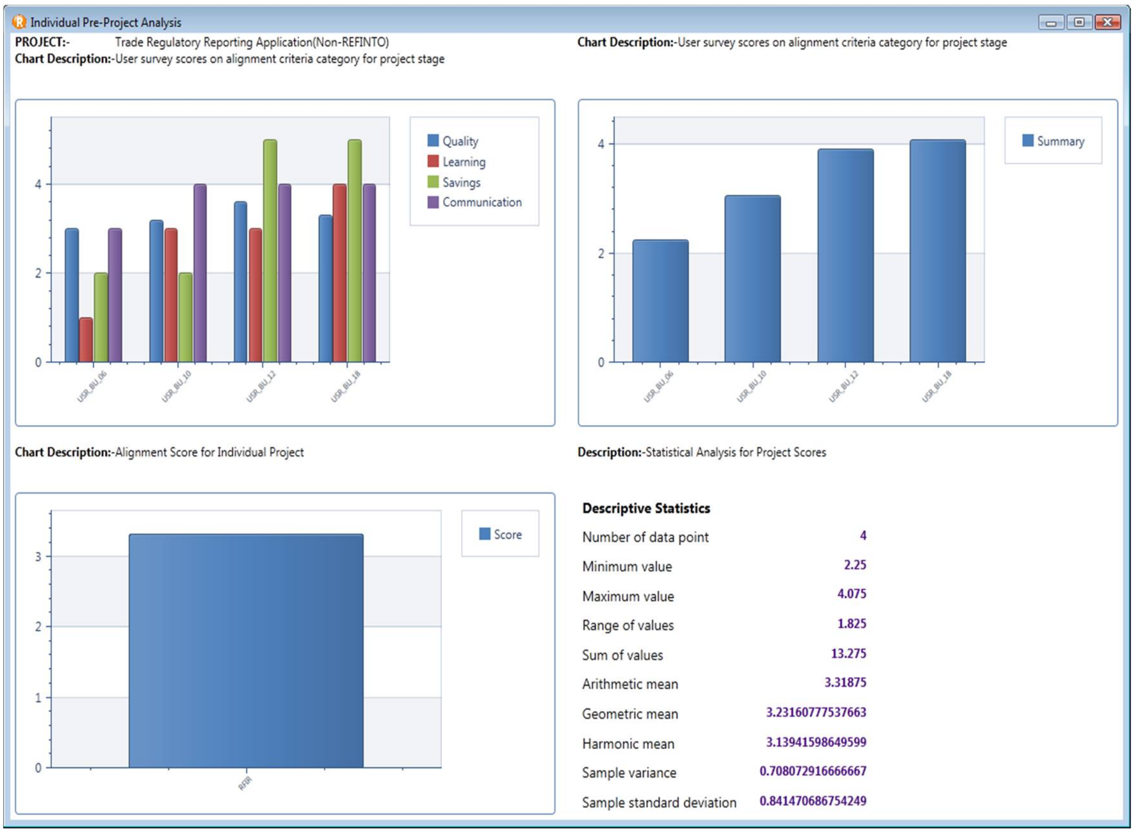

Figure 5.23: Statistical Analysis of a Project in DEMM

5.7 Orthogonal Dimensions of REFINTO Framework and Support Tool

In this section, the orthogonal dimensions and ramifications of using the framework and tool in combination with third party processes or tools is considered from the perspective of what is recommended or supported. It also aimed to highlight how generic and independent the REFINTO framework and support tool are. The matrix of possible scenarios discussed is presented in Table 5.3.

Table 5.3: Orthogonal Dimensions of REFINTO

Combination	REFINTO-led Process	Third Party-led Process
With REFINTO Framework Tool Support	Recommended and supported. Validated in REFINTO & TOOL portfolio in Data Analysis stage	Supported. Not validated in Data Analysis stage. Pre-requisite is mapping third party processes to REFINTO framework processes and RFST features
Without REFINTO Framework Tool Support	Supported. Validated in REFINTO ONLY portfolio in Data Analysis stage. There are limitations like manual matching of requirements and artefact	Supported. Not validated in Data Analysis stage. Pre-requisite is mapping third party processes to REFINTO framework processes
With Third Framework Tool Support	Supported. Not validated in Data Analysis stage. Pre-requisite is mapping REFINTO framework processes to third party processes and tool	Validated in THIRD PARTY portfolio in Data Analysis stage
Without Third Framework Tool Support	Supported. Validated in REFINTO ONLY portfolio in Data Analysis stage. There are limitations like manual matching of requirements and artefact	Validated in THIRD PARTY portfolio in Data Analysis stage

The orthogonal matrix shown in Table 5.3 is described as follows:

I. *REFINTO-led Process with REFINTO Support Tool*. This is the recommended approach to using the REFINTO framework for sustainable business-IT alignment. This scenario is validated under the REFINTO and TOOL portfolio of projects in the data collection, analysis and interpretation chapters of this thesis.

II. *REFINTO-led Process without REFINTO Support Tool*. This scenario is supported and is validated under the REFINTO ONLY portfolio in the data analysis stage. There are limitations such as manual matching of requirements with existing IT artefacts, which is time consuming and error prone. Other benefits of the support tool such as requirement status tracking, version control, and workbook generations.

III. *REFINTO-led Process with Third-Party Support Tool*. This scenario is supported. However, it is not validated in the data analysis stage. The pre-requisite is that third party tool has equivalent or compensating features to those in the REFINTO framework support tool in order to support REFINTO framework processes.

IV. *REFINTO-led Process without Third-Party Support Tool*. This scenario is similar to scenario II and is supported. It is validated under the REFINTO ONLY portfolio in the data analysis stage. The same limitations of scenario II also apply in this scenario.

V. *Third Party-led Process with REFINTO Support Tool.* This scenario is supported provided the requisite mapping of third party processes to the REFINTO framework support tool features is done. It is however not validated in the data analysis stage and presents an opportunity for future studies.

VI. *Third Party-led Process without REFINTO Support Tool*. This scenario has the same limitations of scenario II and is validated in data analysis stage under the THIRD PARTY portfolio.

VII. *Third Party-led Process with Third Party Support Tool*. This scenario involves the use of third party processes and tools, for example MS Project and requirements engineering and management tools, and is validated in the data analysis stage under the THIRD PARTY portfolio.

VIII. *Third Party-led Process without Third Party Support Tool.* This scenario is similar to scenario VI. It is also compared to the REFINTO-led processes in the data analysis stage under the THIRD PARTY portfolio.

In summary, the REFINTO framework and support tool are orthogonal and various combination with or without third party processes and tool support are possible. The REFINTO framework and support tool combination is recommended but it is by no means the only way of extracting value with respect to sustainable business-IT alignment. Third party processes and tools can also be used with the requisite compensating or equivalent features or processes in place to leverage the capabilities and features of the REFINTO framework and support tool.

5.8 Comparison of REFINTO to Other Frameworks

There are other frameworks and tools that aim to support knowledge-based requirement elicitation to ensure high quality requirements. A selection of these tools is compared to the REFINTO framework and support tool to highlight the gaps in practice and academia that contributions made in this study to address the shortcomings.

The tools that are compared are ORE (Kitamura et al., 2009), RECAP (Edwards et al., 1995), REWARD (Scott and Cook, 2003), ElicitO (Al Balushi et al., 2013) and REInDetector (Nguyen et al., 2012). The comparative analysis is provided in Table 5.4.

Enablers of sustainable business-IT alignment were considered when selecting the criteria for the comparison. The criteria include focus on high quality requirements, use of ontologies (or other knowledge-based constructs) for requirement elicitation and management, support for rules and/or case based reasoning, evaluation of requirements quality, provision of business domain lexicon, support for processes in the SDLC beyond requirements elicitation, and focus on business-IT alignment.

Based on the comparison in Table 5.4, it can be argued that REFINTO framework support tool has some advantages over the other frameworks and tools. The focus on business-IT alignment is clearly an advantage. The semi-automated support for requirements management processes like querying and mining, historic tracking, and collaboration are also definitive advantages. It is however also an area of weakness in the REFINTO framework support tool when compared to other tools which offer full elicitation automation. REFINTO framework support tool only supports semi-automation of requirement elicitation.

Table 5.4: Comparison with Other Frameworks

Criteria	REFINTO	ORE	RECAP	REWARD	ElicitO	REInDetector
Reasoning Support	✓	✗	✗	✓	✓	✗
Entire SDLC Support	✓	✗	✗	✗	✗	✗
Multi-User Support	✓	✗	✗	✗	✗	✗
Ontology Support	✓	✓	✗	✓	✓	✓
Partial Req. Management	✓	✗	✗	✗	✗	✗
Quality Checks	✓	✓	✓	✓	✓	✓
Domain Lexicon	✓	✓	✗	✗	✓	✗
Focus on Alignment	✓	✗	✗	✗	✗	✗
Document Parsing	✓	✓	✓	✓	✓	✗
Repository	✓	✗	✓	✓	✗	✗
Full Elicitation Automation	✗	✗	✗	✗	✓	✓
Partial Elicitation Automation	✓	✓	✓	✓	✓	✗
Graphical Interface	✓	✓	✓	✓	✓	✓

5.9 Summary

The processes, framework, and support tool for an ontology-based approach to business-IT alignment through closer interaction of business and IT stakeholders has been presented in this chapter. It builds on the foundation laid in the literature review chapters linking business-IT alignment with requirements engineering and knowledge engineering. The workflow, hypotheses, evaluation and alignment maturity measurement, design, implementation and demonstration perspectives, help to bridge the language and knowledge gaps between stakeholders in a highly specialized business domain.

The framework facilitates a rigorous and structured approach to requirements engineering throughout the software development and project management lifecycles. The antecedents and metrics for gauging business-IT alignment maturity at the tactical and operational levels of organization namely reuse, quality, communication, shared understanding, learning, forecast and estimation, and time and resource saving were discussed under four main groupings – quality, common language, continuous monitoring, service orientation.

The framework support tool, which offers semi-automated support for the framework process has been discussed. The functions of the two modules that make up the tool have been highlighted. The design, architecture and demonstration of the models at the three project stages (pre-project, intra-project, and post-project) have been presented. The domain ontologies that have been developed for recurring functions based on the analysis of historic business requirements in the business domain of study (financial services) have been discussed.

Finally, a comparative analysis of other tools to highlight the gaps that REFINTO framework support tool helps to address has been presented. The scenarios for the use of the REFINTO framework and support tool with third party framework processes and tools have been explored. This chapter forms the basis for the validation of the REFINTO framework and support tool through data analysis discussed in chapters six and seven.

Chapter 6 Data Collection and Analysis

6.1 Introduction

The objective of this chapter is firstly to make comparisons between projects performed with the guidance of the REFINTO framework and support tool to those executed without framework or with a third party framework. Secondly, it is to validate hypotheses underpinning the REFINTO framework. The hypotheses are provided in Table 6.1.

Table 6.1: REFINTO Framework Hypotheses

Code	Hypothesis
H1	The quality of business requirements has an influence on software product quality and together are positively associated with business-IT alignment at the tactical/individual and operational/department levels
H2	Shared knowledge and mutual understanding between business and IT is positively associated with business-IT alignment at the tactical and operational levels
H3	Effective, timely, and frequent communication providing visibility of business process, impact of change, issues (delays, constraints etc.) is positively associated with business-IT alignment at the tactical and operational levels
H4	Learning is positively associated with business-IT alignment at the tactical and operational level
H5	Reuse of asset, knowledge, and requirements lead to time and financial savings, learning and skills. It is positively associated with business-IT alignment at the tactical and operational levels
H6	Continuous evaluation and knowledge gained and codified is positively associated with business-IT alignment at the tactical and operational levels
H7	Accurate forecasting and estimation leads to confidence and mutual trust between business and IT functions and is positively associated with business-IT alignment at the tactical and operational levels
H8	Resource efficiency (cost and time savings) is both a benefit from and an influence on business-IT alignment at the tactical and operational levels

The data collection strategy and instrument used for data collection are discussed. The data collection instrument is summarized in Appendix C. Analysis and breakdown of project participants (respondents to the questionnaires) is provided in Appendix D. Analyses performed on the data collected is presented and the limitations of the data collection and analysis processes are discussed. The study model embodying the eight hypotheses is shown in Figure 6.1. The model includes return on investment (ROI) and productivity, which are influenced by business-IT alignment maturity. This model was used as the basis for factor analysis.

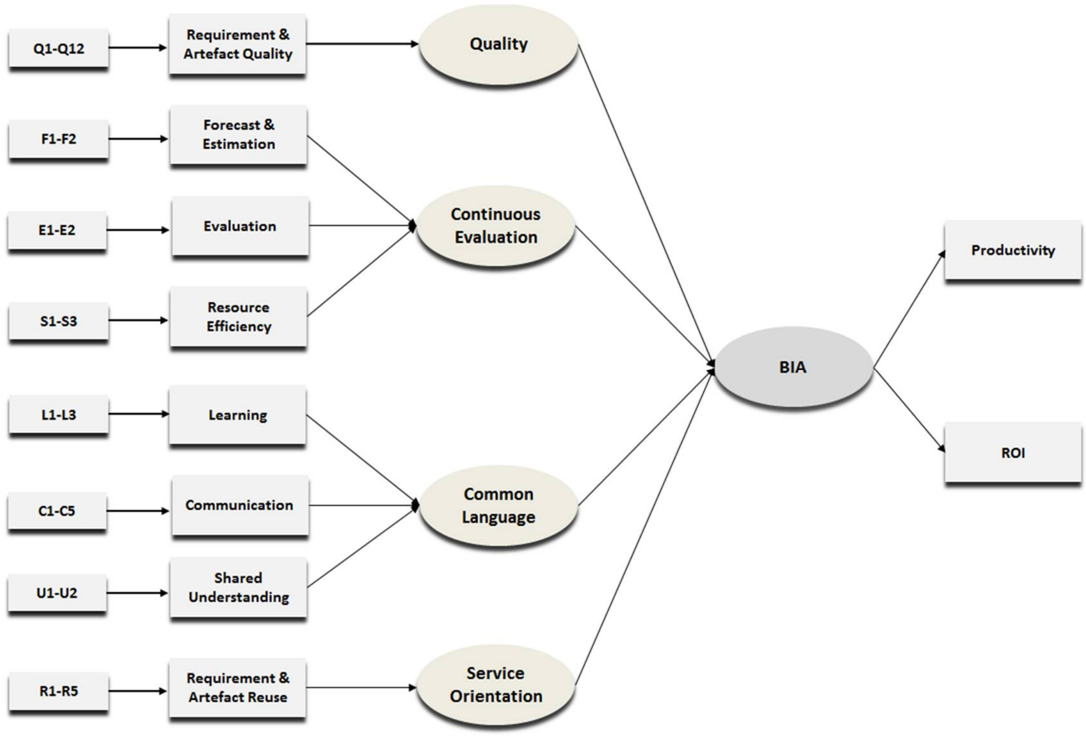

Figure 6.1: Study Model with Instrument Items

6.2 Data Collection Strategy

The data collection strategy adopted in this study evolved over time. It involved collecting data from business-critical projects in the financial services domain. The initial approach (Umoh et al., 2011, Umoh et al., 2012) entailed collecting data from five historic projects that followed ad hoc approaches to requirements elicitation and unstructured software development life cycle (SDLC) processes. No particular attention was paid to business-IT alignment in the course of executing these projects.

The data collected included project plans, build estimates, emails, error and bug lists, testing plans, and survey responses. These were mostly available in the host firm’s issue reporting and tracking systems. Five projects were then executed with the guidance and support of the REFNITO framework and support tool. Data from project plans, build estimates, emails, error and bug lists, testing plans, and user feedback from these projects were also collected. Data analysis was then performed on these two sets of data.

Three limitations were observed with this initial approach. Firstly, it was difficult to determine the impact the support tool had on the alignment scores. To address this limitation, the REFINTO-guided projects data was split into two subsets, one for the REFINTO framework only and the other for the framework and support tool. Secondly, it was observed that there was a bias risk associated with comparison of NON-REFINTO and the two REFINTO project portfolios. To mitigate this risk, another subset containing projects executed with a third party framework was introduced as control. Thirdly, it was adjudged that the number of projects in each subset (five) was small. The number of projects in each subset was doubled to ten to address this limitation. The description of the four portfolios of projects is provided in Table 6.2.

The ‘THIRD PARTY’ portfolio was made up of projects executed using a proprietary framework within the organization this study was carried out, called *BAToolKit.* The framework is based on BABOK[10], PMBOK[11], and ITIL. It has a collection of requirements document templates, project management templates, and is integrated to off-the-shelf workflow and incident management tools. The main driver for this framework is to facilitate communication, notification of status changes, and tracking of project progress.

Table 6.2: REFINTO Framework Project Portfolio

Project Group	Project Portfolio Description	Count
NON-REFINTO	Executed following existing ad hoc or unstructured SDLC approaches	10
REFINTO ONLY	Executed with REFINTO framework without tool support	10
REFINTO & TOOL	Executed with REFINTO framework and tool support	10
THIRD PARTY	Projects executed with third party framework	10

[10] Business Analysis Body of Knowledge (BABOK): http://www.iiba.org/babok-guide.aspx
[11] Project Management Body of Knowledge (PMBOK):http://www.pmi.org/PMBOK-Guide-and-Standards.aspx

To avoid bias, concerted efforts was made to ensure that all projects in the four categorizes were roughly of similar complexities, requiring similar effort, and of equal importance to the business. The challenges encountered with this are discussed later in this chapter.

The alignment score ranges and the interpretations are provided in Table 6.3.

Table 6.3: REFINTO Framework Alignment Scale

Level	Statistical Score	Classification	Range Description
5	4.500-5.000	Optimized Process	Very high quality requirements. Very high quality IT artefact. Highly effective communication. Highly efficient processes reuse of requirements and IT artefacts. Resource efficiency maximized. Domain and process learning significantly improved. Precise estimation actual vs. quoted. Highly effective evaluation mechanisms.
4	3.500-4.499	Improved & Managed Process	High quality requirements. High IT quality artefact. Highly effective communication. Highly efficient processes reuse of requirements and IT artefacts. Improved resource efficiency. Improved domain and process learning. Improved estimation actual vs. quoted. Effective evaluation mechanisms
3	2.500-3.499	Established Focused Process	Requirements of acceptable quality. IT artefact of acceptable quality. Acceptable collaboration and communication. Established processes for reuse of requirements and IT artefacts. Some resource efficiency. Acceptable domain and process learning. Fair estimation actual vs. quoted. Established evaluation mechanisms
2	1.500-2.499	Committed Process	Low requirements quality. Low IT artefact quality. Ineffective collaboration and communication. Inefficient processes reuse of requirements and IT artefacts. Little or no resource efficiency. Minimal domain and process learning. Imprecise estimation actual vs. quoted. Ineffective evaluation mechanisms
1	0-1.499	Initial & Ad Hoc Process	Very low requirements quality. Very low IT artefact quality. Highly ineffective collaboration and communication. Highly inefficient processes reuse of requirements and IT artefacts. Poor resource efficiency. Little or no domain and process learning. Highly imprecise estimation actual vs. quoted. Highly ineffective evaluation mechanisms.

The data collection questionnaires were administered only to project participants. The participants were informed that participation was voluntary and anonymous. Participation in the surveys was positively impacted by management's support for the study. As a result, an overall 94.6% response rate was achieved. The shortfall of 5.4% was due to various factors including vacation, internal redeployment, or terminations.

6.2.1 Data Collection by Project Stage

The processes of data collection for the three stages of each of the projects in the four portfolios and the breakdown of participants at each stage of every project portfolio are provided in Appendix D.

The data for these three stages was collected by administering the data collection instrument specific to the stage being evaluated. Respondents were expected to respond to the request for evaluation preferably at the end of each stage of the project or after all three stages at the post-project stage if not feasible due to operational tasks and pressures. The analysis of the data collected for the three stages is discussed in this chapter.

6.2.2 Data Collection by Project Portfolio

The breakdown of project participants in each project portfolio provided in Appendix D gives an indication of the size of the data samples collected. The process of data collection for each of the four portfolios of projects is provided in the appendix. The details of the projects, the data collection instrument administration for the portfolio are covered. The classification of project was by the main function the application performs. In practical terms however, a project may have other functions in addition to its main function. For instance, a reporting project may have a reconciliation function.

6.2.3 Summary of Data Collection

Projects in the four categories were selected based on criteria such as similarity in functionality, complexity, and size. This was done to ensure objective comparison. The summary of the data collection by stage and project classification and the total number of samples available for data analysis is provided in Table 6.4.

The sample size for each stage is over 200. The average sample size for the project portfolios is 158. This satisfies the criteria for recommended sample size for a SEM study (Lei and Wu, 2007, Garver and Mentzer, 1999, Hoelter, 1993).

Table 6.4: Data Collection Summary

Stage	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY	Total
Pre-Project	51	57	53	48	209
Intra-Project	53	57	54	49	213
Post-Project	52	57	54	49	212
Total	156	171	161	146	634

The recommended sample size for a study using SEM is generally ten times the number of items in the data collection instrument (Ullman, 2006, Mueller and Hancock, 2008, McDonald and Ho, 2002). In this study, that would be 340 (10 * 34 = 340). The sample size of data collected for this study satisfies this criterion.

6.3 Data Collection Instrument

The data collection instrument, a 34-item closed form questionnaire, was based on the hypotheses and metrics identified for measuring the factors that make up the hypotheses, already discussed in preceding chapters.

The design of the data collection instrument was guided by questionnaire design guidelines for information systems. The Instrument is presented in Appendix C. Data collection with the instrument was based on role, projects participated in, and project stage. Personal information of respondents and proprietary information relating to the host organization where the study was conducted is anonymized.

Actions taken to avoid bias during data collection include targeted project portfolio composition and project execution timing. Projects of similar complexity and size and requiring similar resources were allocated to each project portfolio. The project execution was concurrent in many cases, driven by business priority. It was however difficult to assign personnel to the same project portfolio consistently throughout the study. Introduction of the THIRD PARTY project portfolio was also an important bias mitigation action.

6.4 Data Analysis

Analyses performed on the samples for the three stages of projects in the four portfolios of projects are presented in this section. The analyses were performed in line with the objectives of the data collection and analysis stated earlier. These were to compare the four project portfolios and validate the REFINTO framework hypotheses using the study model. The comparison of projects in the four project portfolios was done using descriptive statistics while the validation of the study model and hypotheses was performed with factor analysis.

Whereas descriptive statistics performed on the data was straightforward, factor analysis was challenging. The primary tool used for factor analysis was LISREL version 9.1.0 and following best practices for SEM made up of four steps. These are model conceptualization or specification and hypotheses postulation, parameter identification and estimation, model evaluation and data-model fit assessment, and model modification (Ullman, 2006, Mueller and Hancock, 2008, McDonald and Ho, 2002). These steps were followed for data analyses performed in this study.

6.4.1 Data Analysis by Project Portfolio

Descriptive statistics for the consolidated data collected for each of the four project portfolios by administering the data collection instrument is shown in Table 6.5. The fully consolidated data for the four portfolios from all participants is provided in Table 6.6.

Table 6.5: Descriptive Statistics for Consolidate Portfolio Data

	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Mean	2.5959	3.5782	3.7215	3.2318
Standard Error	0.0751	0.0317	0.0316	0.0702
Median	2.7035	3.5655	3.766	3.2235
Standard Deviation	0.4378	0.1850	0.1842	0.4091
Sample Variance	0.1916	0.0342	0.0339	0.1674
Range	1.692	0.717	0.7	2.015
Minimum	1.902	3.139	3.294	1.929
Maximum	3.594	3.856	3.994	3.944
Sum	88.259	121.66	126.531	109.881
Count	34	34	34	34

The graphical representation of the consolidated data is depicted in Figure 6.2.

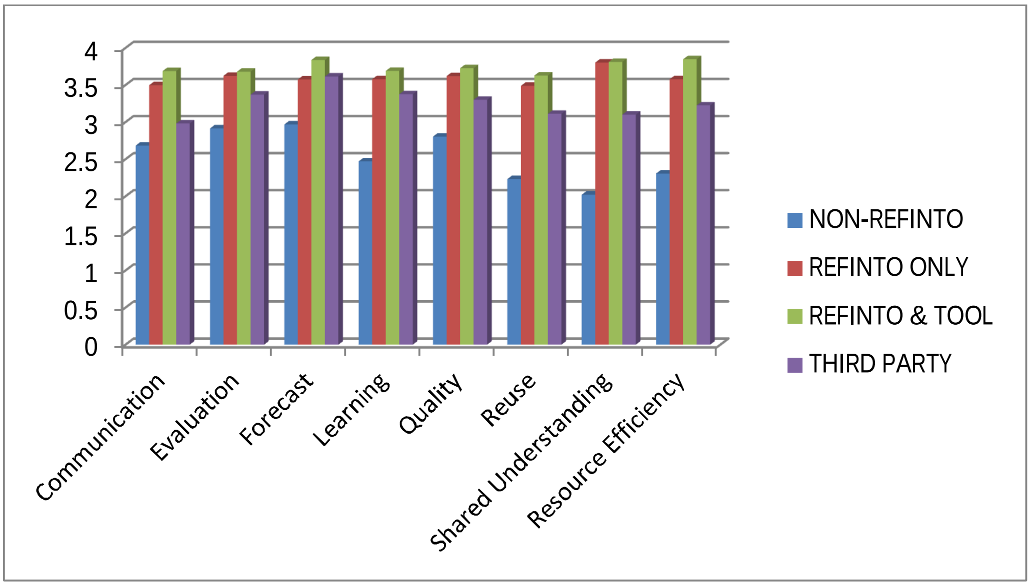

Figure 6.2: Graphical Analysis for Project Portfolios

The data in Table 6.6 shows the aggregated alignment maturity scores of all projects for each project portfolio. The graphical representation shows ranking of the project portfolios in descending order as REFINTO & TOOL, REFINTO ONLY, THIRD PARTY, and NON-REFINTO. Further breakdown of the data by portfolio which provides better insight into the data is explored in this section.

Table 6.6: Consolidated Data for Project Portfolios

Construct	Stage	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY	Variance	Std. Dev.
C1	Pre-Project	1.992	3.628	3.615	3.186	0.067	0.259
L1	Pre-Project	2.292	3.502	3.534	3.296	0.076	0.275
Q1	Pre-Project	3.04	3.562	3.44	3.132	0.081	0.285
Q2	Pre-Project	2.844	3.548	3.294	3.104	0.098	0.313
Q3	Pre-Project	2.828	3.258	3.889	3.64	0.051	0.225
Q4	Pre-Project	2.83	3.372	3.414	3.607	0.116	0.34
Q5	Pre-Project	3.108	3.822	3.921	3.944	0.238	0.488
Q6	Pre-Project	2.692	3.61	3.77	3.01	0.268	0.518
Q7	Pre-Project	3.594	3.844	3.885	3.118	0.435	0.66
Q8	Pre-Project	2.844	3.762	3.895	2.986	0.628	0.793
Q9	Pre-Project	2.828	3.534	3.669	3.214	0.289	0.537
Q10	Pre-Project	2.462	3.856	3.797	3.291	0.098	0.313
Q11	Pre-Project	2.265	3.612	3.782	3.104	0.051	0.225
R1	Pre-Project	1.902	3.511	3.731	3.829	0.554	0.744
S1	Pre-Project	2.152	3.842	3.994	3.285	0.582	0.763
U1	Pre-Project	2.054	3.807	3.879	3.184	0.574	0.758
U2	Pre-Project	1.989	3.798	3.815	3.018	0.467	0.683
C2	Intra-Project	2.208	3.528	3.818	1.988	0.067	0.835
L2	Intra-Project	2.83	3.566	3.574	3.723	0.246	0.496
R2	Intra-Project	2.046	3.139	3.631	1.929	0.551	0.742
R3	Intra-Project	2.484	3.456	3.798	3.407	0.121	0.348
R4	Intra-Project	1.956	3.296	3.774	3.304	0.33	0.575
S2	Intra-Project	2.778	3.261	3.762	3.002	0.134	0.366
C3	Post-Project	2.701	3.223	3.615	3.112	0.051	0.227
C4	Post-Project	3.522	3.454	3.503	3.209	0.075	0.275
C5	Post-Project	2.706	3.656	3.892	3.411	0.16	0.4
E1	Post-Project	3.02	3.69	3.702	3.233	0.069	0.263
E2	Post-Project	2.809	3.558	3.657	3.508	0.146	0.382
F1	Post-Project	2.816	3.494	3.889	3.596	0.025	0.157
F2	Post-Project	3.121	3.656	3.785	3.634	0.185	0.431
L3	Post-Project	2.291	3.668	3.962	3.109	0.216	0.464
Q12	Post-Project	2.521	3.658	3.958	2.983	0.317	0.563
R5	Post-Project	2.774	3.565	3.686	3.094	0.3	0.547
S3	Post-Project	2.152	3.446	3.679	3.388	0.355	0.596

6.4.1.1 Data Analysis for NON-REFINTO

The summary of alignment scores obtained from data collected for NON-REFINTO projects is provided in Table 6.7. The descriptive statistics of the data is provided in Table 6.8 and in chart format, as shown in Figure 6.3.

Table 6.7: Alignment Scores for NON-REFINTO Portfolio

Construct	Score
Communication	2.6837
Evaluation	2.9145
Forecast	2.9685
Learning	2.4713
Quality	2.8053
Reuse	2.2324
Shared Understanding	2.0215
Resource Efficiency	2.3067

Table 6.8: Descriptive Statistics for NON-REFINTO Portfolio

	NON-REFINTO
Mean	2.5959
Standard Error	0.0751
Median	2.7035
Standard Deviation	0.4378
Sample Variance	0.1917
Range	1.692
Minimum	1.902
Maximum	3.594
Sum	88.259
Count	34

The alignment score for the pre-project stage was low (2.596) indicating a borderline established focused process. This indicates problems with the requirements elicitation process. Other factors also have low scores. For example, shared understanding had the lowest score at 2.0215 and reuse (2.2324) was equally low. Forecast and evaluation were the highest scores at 2.9685 and 2.9145 respectively.

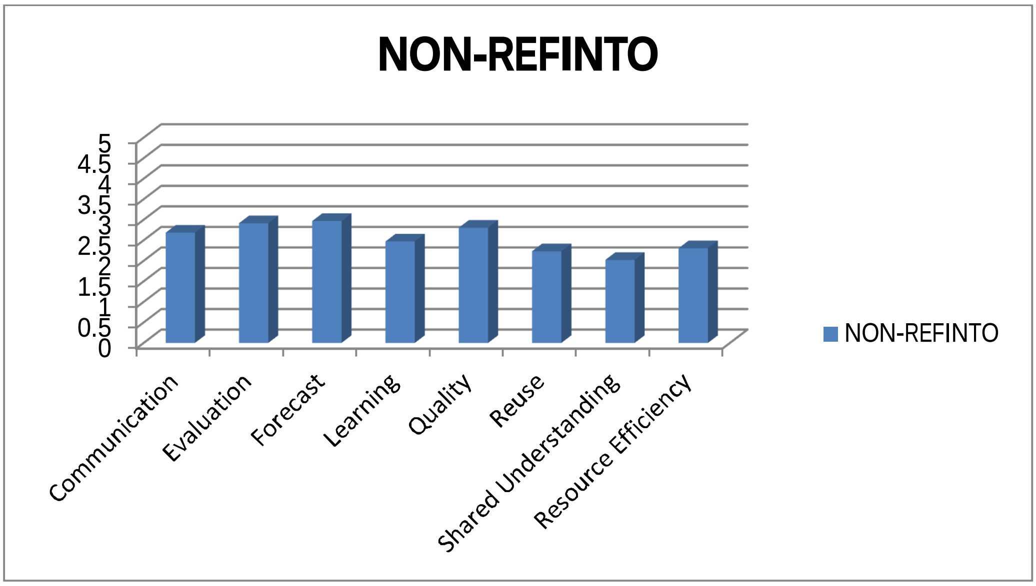

Figure 6.3: Graphical Analysis for NON-REFINTO Portfolio

6.4.1.2 Data Analysis for REFINTO ONLY

The summary of alignment scores obtained from data collected for REFINTO ONLY projects is provided in Table 6.9. Descriptive statistics for this portfolio is shown in Table 6.10 and in chart format, depicted in Figure 6.4.

The mean alignment score (3.5782) indicates better alignment in this portfolio of projects compared to the NON-REFINTO portfolio. This can be attributed to the structured and relatively more disciplined process followed at requirements elicitation and the collaboration between business and IT stakeholders. This reflected in the shared understanding (3.8025) and quality (3.6198) scores.

The score for reuse (3.489) was the lowest score followed by communication (3.4978). This can be attributed to the manual processes with no tool support used for matching requirements to artefacts matching, an important factor in the *elicit-refine-persist* process. It was also much harder to provide accurate estimates and convey progress updates to business stakeholders thereby impeding visibility of the implementation process.

Table 6.9: Alignment Scores for REFINTO ONLY Portfolio

Construct	Score
Communication	3.4978
Evaluation	3.624
Forecast	3.575
Learning	3.5789
Quality	3.6198
Reuse	3.489
Shared Understanding	3.8025
Resource Efficiency	3.5772

Table 6.10: Descriptive Statistics for REFINTO ONLY Portfolio

	REFINTO ONLY
Mean	3.5782
Standard Error	0.0317
Median	3.5655
Standard Deviation	0.1850
Sample Variance	0.0342
Range	0.717
Minimum	3.139
Maximum	3.856
Sum	121.66
Count	34

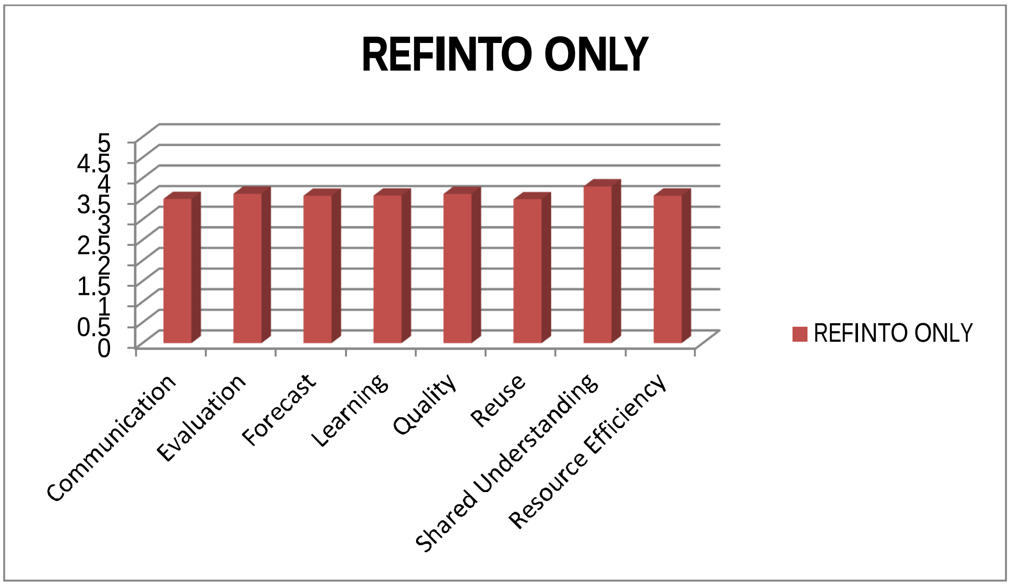

Figure 6.4: Graphical Analysis for REFINTO ONLY Portfolio

6.4.1.3 Data Analysis for REFINTO & TOOL

The summary of alignment scores obtained from data collected for REFIINTO and TOOL portfolio of projects is provided in Table 6.11. The descriptive statistics analysis of the data is provided in Table 6.12 and alignment scores for the portfolio is depicted in Figure 6.5.

Table 6.11: Alignment Scores for REFINTO & TOOL Portfolio

Construct	Score
Communication	3.6886
Evaluation	3.6795
Forecast	3.837
Learning	3.69
Quality	3.7262
Reuse	3.6284
Shared Understanding	3.8117
Resource Efficiency	3.847

Table 6.12: Descriptive Statistics for REFINTO & TOOL Portfolio

	REFINTO & TOOL
Mean	3.7215
Standard Error	0.0316
Median	3.766
Standard Deviation	0.1842
Sample Variance	0.0339
Range	0.7
Minimum	3.294
Maximum	3.994
Sum	126.531
Count	34

The mean score for this portfolio (3.7215) indicates very good alignment (Improved/Managed Process). This can be attributed to the collaborative, structured, disciplined, and tool supported process followed for requirements elicitation, project execution, and collaborative review process.

The scores for shared understanding (3.8117) and quality (3.7262) reflected this. The scores for reuse (3.6284) and communication (3.6886) were relatively better than those of the REFINTO ONLY portfolio. This can be attributed to the semi-automated (with tool support) requirements to historic cases/artefacts matching which facilitates the *elicit-refine-persist* and *ontology-requirement-artefact* processes. With tool support, it was easier to communicate project estimates and report progress to business stakeholders. This was reflected in the better communication and forecast (3.837) scores. The saving scores (3.847) obtained was the highest score. This implies that time and cost savings are made through the structured and rigorous processes. Interestingly, the reuse score (3.6284) was the lowest, however, when compared to the REFINTO ONLY (3.489) and NON-REFINTO (2.2324), it showed marked improvement.

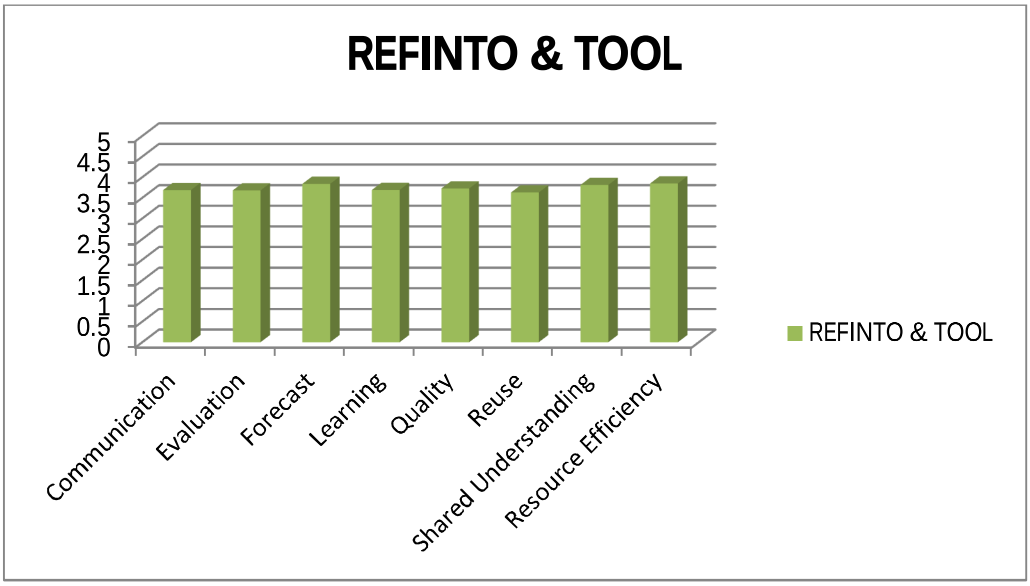

Figure 6.5: Graphical Analysis for REFINTO & TOOL Portfolio

6.4.1.4 Data Analysis for THIRD PARTY

The summary of alignment scores obtained from data collected for THIRD PARTY projects is provided in Table 6.13. The descriptive statistics analysis of the data is provided in Table 6.14. The chart form of the THIRD PARTY portfolio data is depicted in Figure 6.6.

Table 6.13: Alignment Scores for THIRD PARTY Portfolio

Construct	Score
Communication	2.9812
Evaluation	3.3705
Forecast	3.615
Learning	3.376
Quality	3.3004
Reuse	3.1126
Shared Understanding	3.101
Resource Efficiency	3.225

Table 6.14: Descriptive Statistics for THIRD PARTY Portfolio

	THIRD PARTY
Mean	3.2318
Standard Error	0.0702
Median	3.2235
Standard Deviation	0.4091
Sample Variance	0.1674
Range	2.015
Minimum	1.929
Maximum	3.944
Sum	109.881
Count	34

The mean score for this portfolio (3.2318) indicated marked improved alignment (Established Focused Process) compared to the NON-REFINTO portfolio (2.5959). In comparison to the two REFINTO portfolios (3.5782 for REFINTO ONLY and 3.7215 for REFINTO and TOOL) was less impressive. This can be attributed to the structured and more disciplined process followed at requirements elicitation and implementation stages.

The scores for reuse (3.1126) and communication (2.9812) were markedly better than those of the NON-REFINTO portfolio but less than those for the REFINTO portfolios. The highest scores in this portfolio were forecast (3.615) and learning (3.376). This underscores the importance of following structured and rigorous processes towards improving addressing business-IT alignment maturity.

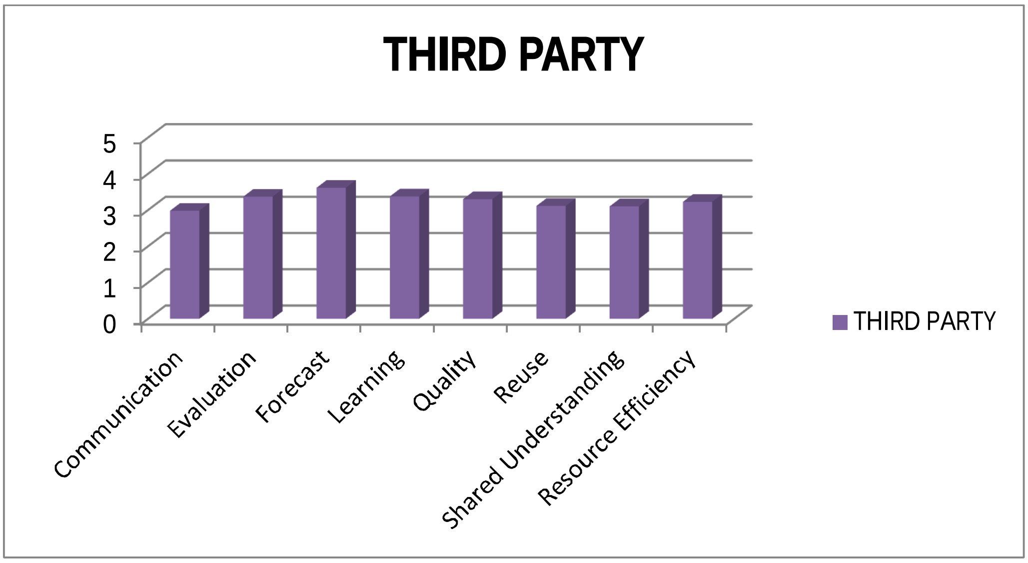

Figure 6.6: Graphical Analysis for THIRD PARTY Portfolio

6.4.2 Data Analysis by Stage

The consolidated data were further analysed by the three project stages. This provided gainful insight on alignment through the different stages and if there was correlation between alignment scores at the three stages.

6.4.2.1 Data Analysis for Pre-Project Stage

The summary of alignment scores obtained from data collected for the pre-project stage for all project portfolios is provided in Table 6.15. The descriptive statistics analysis of the data is provided in Table 6.16. The pre-project alignment scores are shown in Figure 6.7. The alignment scores for the pre-project stage were improved with the introduction of structure and rigour into the elicitation process. This was reflected in the improvement in quality from 2.8312 (Established Focused Process in NON-REFINTO) to 3.6674 (Improved/Managed Process in REFINTO & TOOL). Communication has improved from 1.992 (Committed Process in NON-REFINTO) to 3.731 (Improved/Managed Process in REFINTO & TOOL) and shared understanding from 2.0215 (Committed Process in NON-REFINTO) compared to 3.9365 (Improved/Managed Process in REFINTO & TOOL).

Learning and resource efficiency scores were also significantly improved. This indicates strong support for the argument that a tool to facilitate elicit-refine-persist contributes to better pre-project alignment scores. This was reflected in the significant improvement in reuse which was 1.902 (Committed Process) in NON-REFINTO compared to 3.895 (Improved/Managed Process) in REFINTO & TOOL.

Table 6.15: Pre-Project Alignment Scores

Construct	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Communication	1.992	3.511	3.731	3.829
Learning	2.292	3.534	3.669	3.214
Quality	2.8312	3.6013	3.6674	3.3395
Reuse	1.902	3.762	3.895	2.986
Shared Understanding	2.0215	3.8245	3.9365	3.2345
Resource Efficiency	2.152	3.798	3.815	3.018

Table 6.16: Pre-Project Alignment Descriptive Statistics

	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Mean	2.5602	3.6393	3.7249	3.3089
Standard Error	0.1148	0.0433	0.0489	0.0694
Median	2.692	3.612	3.782	3.214
Standard Deviation	0.4732	0.1787	0.1842	0.2015
Sample Variance	0.2239	0.0319	0.0406	0.0819
Range	1.692	0.598	0.7	0.958
Minimum	1.902	3.258	3.294	2.986
Maximum	3.594	3.856	3.994	3.944
Sum	43.524	61.868	63.324	56.251
Count	17	17	17	17

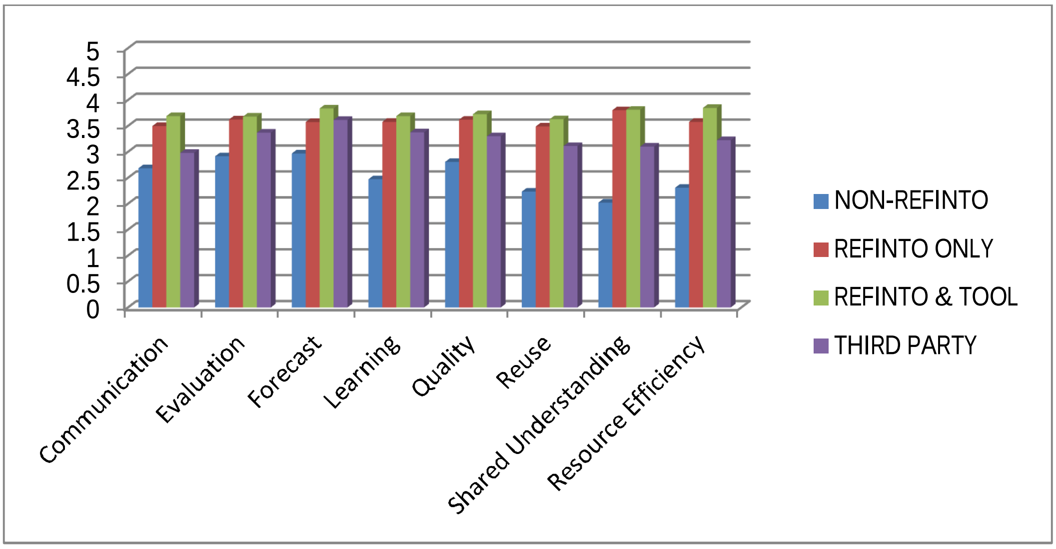

Figure 6.7: Pre-Project Alignment for all Project Portfolios

6.4.2.2 Data Analysis for Intra-Project Stage

The summary of alignment scores obtained from data collected for the intra-project stage for all project portfolios is provided in Table 6.17. The descriptive statistics analysis of the data is provided in Table 6.18. The alignment scores for the intra-project stage is depicted in Figure 6.8.

Table 6.17: Intra-Project Alignment

Construct	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Communication	2.208	3.296	3.774	3.304
Learning	2.83	3.456	3.798	3.407
Reuse	2.162	3.6743	3.6743	2.5467
Resource Efficiency	2.778	3.798	3.762	3.002

Table 6.18: Intra-Project Alignment

	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Mean	2.3837	3.3857	3.7149	2.8921
Standard Error	0.1284	0.0591	0.0361	0.2620
Median	2.384	3.454	3.762	3.002
Standard Deviation	0.3397	0.1564	0.0955	0.6931
Sample Variance	0.1154	-1.0948	0.0091	0.4804
Range	0.874	0.427	0.244	1.794
Minimum	1.956	3.139	3.574	1.929
Maximum	2.83	3.566	3.818	3.723
Sum	16.686	23.7	26.004	20.245
Count	7	7	7	7

The most significant improvement in alignment scores observed was reuse, which had gone from 2.162 (Committed Process in NON-REFINTO) to 3.6743(Improved/Managed Process in REFINTO & TOOL). This can be attributed to the semi-automated matching of requirements and artefacts, which facilitates reuse. Communication and reuse also moved from Committed Process in NON-REFINTO to Improved/Managed Process in REFINTO & TOOL and to a lesser extent (Established Focused Process) in REFINTO ONLY. Learning and resource efficiency also improved from Established Focused Process in NON-REFINTO to Improved/Managed Process in REFINTO & TOOL.

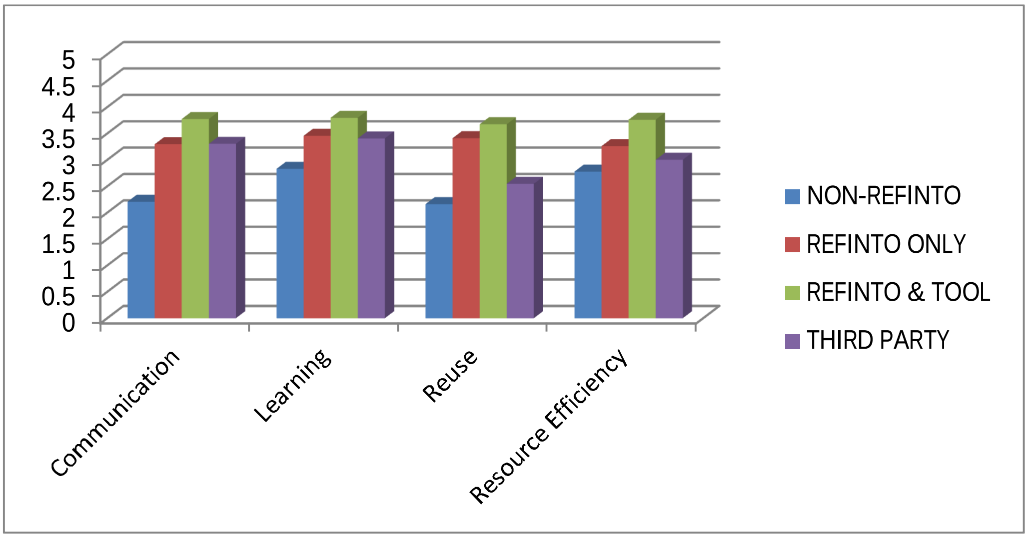

Figure 6.8: Intra-Project Alignment for all Project Portfolios

6.4.2.3 Data Analysis for Post-Project Stage

The summary of alignment scores obtained from analysis of data collected for the post-project stage for all project portfolios is provided in Table 6.19. The descriptive statistics analysis of the data is provided in Table 6.20. The post-project alignment scores are depicted in Figure 6.9.

The post-project stage outcomes were found to depend to some extent on the pre-project and intra-project stages processes such as the quality of requirements elicited, matching requirements with artefacts, reuse of historic requirements/assets, estimation of project estimates, and communication with stakeholders. The forums held for requirements elicitation, development, project management, and evaluation seemed to have positive impacts on alignment scores at the post-project stage for the REFINTO portfolios.

Table 6.19: Post-Project Alignment

Construct	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Communication	3.1115	3.526	3.773	3.552
Evaluation	2.9145	3.3385	3.559	3.1605
Forecast	2.9685	3.673	3.797	3.322
Learning	2.292	3.565	3.686	3.094
Reuse	2.774	3.668	3.962	3.109
Resource Efficiency	2.152	3.446	3.679	3.388

Table 6.20: Post-Project Alignment

	NON-REFINTO	REFINTO ONLY	REFINTO & TOOL	THIRD PARTY
Mean	2.7667	3.5516	3.7571	3.2979
Standard Error	0.1143	0.04243	0.0455	0.0667
Median	2.774	3.565	3.702	3.233
Standard Deviation	0.3789	0.1407	0.1509	0.2214
Sample Variance	0.1436	0.0198	0.0228	0.0490
Range	1.37	0.467	0.459	0.651
Minimum	2.152	3.223	3.503	2.983
Maximum	3.522	3.69	3.962	3.634
Sum	30.434	39.068	41.328	36.277
Count	11	11	11	11

The most significant improvement in alignment scores observed was reuse, which had gone from 2.774 (Established Focused Process in NON-REFINTO) to 3.962 (Improved/Managed Process in REFINTO & TOOL). Linked to this were improvements in forecast, resource efficiency, and learning. Forecast improved from 2.9685 (Established Focused Process in NON-REFINTO) to 3.797 (Improved/Managed Process in REFINTO & TOOL) and to a lesser extent (3.322) in the THIRD PARTY portfolio. Resource Efficiency scores improved from 2.152 (Committed Process in NON-REFINTO) to 3.679 (Improved/Managed Process in REFINTO & TOOL). The resource efficiency scores in both THIRD PARTY and REFINTO ONLY were better than the NON-REFINTO portfolio, at Established Focused Process maturity levels.

Other alignment scores showed improvements using structured and rigorous processes. For example, learning scores improved from Committed Process in NON-REFINTO (2.292) to Improved/Managed Process in both REFINTO portfolios and Established Focused Process in the THIRD PARTY portfolio. Communication scores improved from 3.115 (Established Focused Process in NON-REFINTO) to Improved/Managed Process in REFINTO ONLY, REFINTO & TOOL and THIRD PARTY.

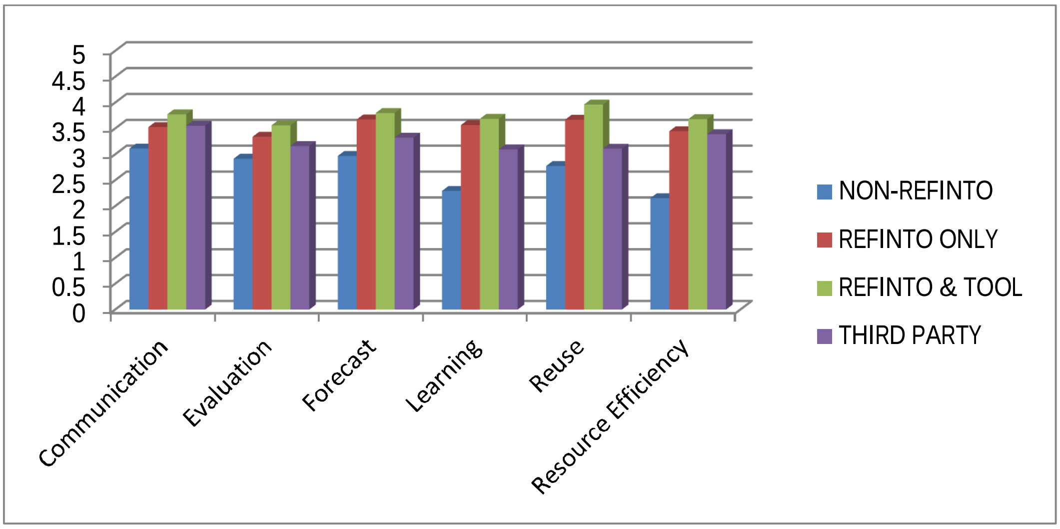

Figure 6.9: Post-Project Alignment for all Project Portfolios

6.4.3 Factor Analysis for Data

Factor analysis, an aspect of SEM was introduced in chapter four. The four steps in a SEM study were identified namely model conceptualization, model estimation, model evaluation and fit assessment, and model modification. This section covers the use of factor analysis for construct validity, criterion-related validity, reliability, and item-total correlation (Ullman, 2006, Lei and Wu, 2007).

Model Conceptualization/Specification

In chapter five the theoretical foundations of the hypotheses underpinning the REFINTO framework, were discussed. These discussions were the basis of the hypotheses model depicted in Figure 6.1. This fulfils the model conceptualization step of a SEM study. It also implies that the most appropriate SEM technique for the other three steps was confirmatory factor analysis (CFA). There are two types of CFA namely, individual CFA and pooled CFA. Individual CFA involves running latent variables one by one to achieve required model fitness whereas pooled CFA involves running the variables simultaneously. Pooled CFA is quicker and more efficient (Chong et al., 2014) and was used in this study.

It was also highlighted that exploratory factor analysis (EFA) and Principal Component Analysis (PCA) are performed when factor structure are not clear. This notwithstanding, exploratory factor analysis was performed with Varimax rotation, Kaiser Normalization, and eigenvalues greater than 1.0 following extant literature (Kearns and Lederer, 2000). The output was identical to the theorized structure conceptualized a priori in the model depicted in Figure 6.1. Following this validation, CFA using SEM was performed. This approach is followed in other studies using factor analysis for alignment hypotheses validation (Kearns and Lederer, 2000, Luftman et al., 2008, Charoensuk et al., 2014).

Model Estimation

CFA was performed on the data using SEM through LISREL version 9.1.0 (SIMPLIS approach) which allowed for simultaneous estimation of the 34 variables as well as indirect, direct and total association (Garver and Mentzer, 1999, Hoelter, 1993). The codes used are defined in Table 6.21.

Table 6.21: Code Definitions for Standardized Estimation Coefficients

Factor/Item	Code
Communication	COMM
Evaluation	EVAL
Forecast & Estimation	FORE
Learning	LEARN
Quality	QUAL/QUALITY
Reuse	REUSE
Shared Understanding	UND
Resource Efficiency	SAVE
Service Orientation	SERVORNT
Common Language	COMMLANG
Continuous Monitoring	CONMONT
Business-IT Alignment	BIA
Productivity	PRODTVTY
Return on investment	ROI

Model Fit Assessment

The result of correlation assessment performed is presented in Table 6.22 showing all the question coefficients that are above 0.5. The cut-off threshold adopted was 0.5 following Doll and Torkzadeh (Doll and Torkzadeh, 1988). The cut-off threshold for the item-criterion correlation was set to 0.4. The result of the item-criterion correlation (Cote et al., 2001) performed is presented in Table 6.23 showing the factor matrix of the 34 items on the data collection instrument.

All factors in the instrument had correlation coefficients greater than 0.4. The factor loading criterion chosen for this study was 0.5. All eight factors are well above 0.5 and were retained in the model. A summarized rotated factor matrix is shown in Table 6.24. The model fitting criteria and the results for this model is presented and discussed in chapter seven.

Table 6.22: Item-Total Correlation

Factor	Correlation	Coefficient Alpha
COMM1	0.645	<.0001
COMM2	0.679	<.0001
COMM3	0.681	<.0001
COMM4	0.567	<.0001
COMM5	0.513	<.0001
EVAL1	0.684	<.0001
EVAL2	0.522	<.0001
FORE1	0.665	<.0001
FORE2	0.566	<.0001
LEARN1	0.524	<.0001
LEARN2	0.598	<.0001
LEARN3	0.552	<.0001
QUAL1	0.681	<.0001
QUAL2	0.623	<.0001
QUAL3	0.576	<.0001
QUAL4	0.602	<.0001
QUAL5	0.575	<.0001
QUAL6	0.608	<.0001
QUAL7	0.573	<.0001
QUAL8	0.656	<.0001
QUAL9	0.586	<.0001
QUAL10	0.606	<.0001
QUAL11	0.611	<.0001
QUAL12	0.676	<.0001
REUSE1	0.589	<.0001
REUSE2	0.544	<.0001
REUSE3	0.596	<.0001
REUSE4	0.678	<.0001
REUSE5	0.622	<.0001
SAVE1	0.613	<.0001
SAVE2	0.564	<.0001
SAVE3	0.681	<.0001
UND1	0.542	<.0001
UND2	0.502	<.0001

The maximum likelihood method was used to assess the model. Convergence reliability and validity of alignment scores were evaluated by assessing the adjustment levels of the model and the causality coefficient linking business-IT alignment and return on investment and productivity.

Table 6.23: Inter-correlations Matrix of Study Variables for Research Data

	Quality	Shared Understanding	Communication	Learning	Reuse	Evaluation	Forecast & Estimate	Resource Efficiency
Q1	**0.84**	0.43	0.25	0.4	0.41	0.35	0.34	0.42
Q2	**0.79**	0.44	0.45	0.33	0.37	0.3	0.34	0.32
Q3	**0.82**	0.39	0.25	0.36	0.31	0.31	0.36	0.37
Q4	**0.8**	0.43	0.46	0.43	0.43	0.42	0.43	0.4
Q5	**0.82**	0.22	0.38	0.32	0.34	0.32	0.35	0.32
Q6	**0.81**	0.41	0.39	0.34	0.38	0.33	0.38	0.3
Q7	**0.75**	0.5	0.4	0.4	0.32	0.38	0.44	0.42
Q8	**0.83**	0.54	0.56	0.53	0.5	0.54	0.52	0.51
Q9	**0.73**	0.5	0.52	0.53	0.47	0.49	0.41	0.41
Q10	**0.88**	0.56	0.49	0.44	0.4	0.42	0.43	0.43
Q11	**0.87**	0.65	0.62	0.59	0.52	0.54	0.59	0.4
Q12	**0.82**	0.42	0.41	0.52	0.44	0.47	0.33	0.48
U1	0.41	**0.87**	0.57	0.54	0.52	0.53	0.21	0.42
U2	0.38	**0.81**	0.61	0.61	0.53	0.54	0.24	0.45
C1	0.52	0.53	**0.89**	0.51	0.54	0.51	0.33	0.37
C2	0.41	0.63	**0.86**	0.42	0.41	0.42	0.42	0.32
C3	0.46	0.62	**0.87**	0.42	0.44	0.42	0.4	0.4
C4	0.41	0.61	**0.87**	0.48	0.42	0.39	0.44	0.41
C5	0.44	0.59	**0.86**	0.45	0.43	0.38	0.42	0.39
L1	0.39	0.55	0.5	**0.9**	0.44	0.45	0.41	0.1
L2	0.3	0.51	0.52	**0.81**	0.41	0.41	0.42	0.33
L3	0.41	0.47	0.47	**0.81**	0.41	0.48	0.46	0.29
R1	0.64	0.43	0.45	0.63	**0.81**	0.43	0.42	0.42
R2	0.58	0.5	0.46	0.45	**0.81**	0.4	0.42	0.43
R3	0.57	0.51	0.32	0.55	**0.84**	0.41	0.41	0.41
R4	0.53	0.52	0.29	0.47	**0.85**	0.41	0.41	0.42
R5	0.6	0.48	0.25	0.48	**0.84**	0.41	0.42	0.42
E1	0.29	0.24	0.22	0.42	0.42	**0.87**	0.42	0.42
E2	0.23	0.26	0.21	0.35	0.4	**0.86**	0.51	0.53
F1	0.44	0.41	0.64	0.29	0.66	0.66	**0.85**	0.48
F2	0.43	0.38	0.56	0.22	0.59	0.47	**0.8**	0.41
S1	0.57	0.54	0.45	0.51	0.59	0.51	0.44	**0.84**
S2	0.45	0.5	0.42	0.42	0.53	0.42	0.43	**0.85**
S3	0.52	0.47	0.43	0.46	0.47	0.36	0.44	**0.84**

Table 6.24: Inter-correlations of study variables of Research Data

	Communication	Reuse	Shared Understanding	Learning	Evaluation	Forecast & Estimate	Quality	Resource Efficiency
Communication	**0.87**							
Reuse	0.46	**0.84**						
Shared Understanding	0.61	0.53	**0.84**					
Learning	0.52	0.44	0.55	**0.84**				
Evaluation	0.22	0.42	0.26	0.48	**0.87**			
Forecast & Estimate	0.64	0.66	0.41	0.46	0.66	**0.83**		
Quality	0.52	0.6	0.65	0.29	0.54	0.57	**0.81**	
Resource Efficiency	0.45	0.59	0.54	0.51	0.51	0.44	0.51	**0.84**

Table 6.25: Summary of Standardized Estimation Coefficients

Standardized Estimation Coefficients	Estimator
COMMLANG ← BIA	0.695
CONTMONG ← BIA	0.553
SERVORNT ← BIA	0.641
QUALITY← BIA	0.751
ROI ← BIA	0.633
PRODVTY ← BIA	0.584
UND ← COMMLANG	0.681
LEARN ← COMMLANG	0.773
COMM ← COMMLANG	0.732
EVAL ← CONTMONG	0.624
FORE ← CONTMONG	0.661
SAVE ← CONTMONG	0.744
QUAL ← QUALITY	0.781
REUSE ← SERVORNT	0.752

Model Evaluation and Modification

The verification of the model with sampling data is in two parts, the structural and measurement models. The discriminant validity checks performed on the data does not indicate any evidence of multidimensionality. This implies that the exogenous variables are not highly correlated. Model modification was not feasible due to organizational constraints, however, suggestions for modification are proposed. The study model shown in Figure 6.10 was implemented in LISREL 9.10 using the SIMPLIS approach.

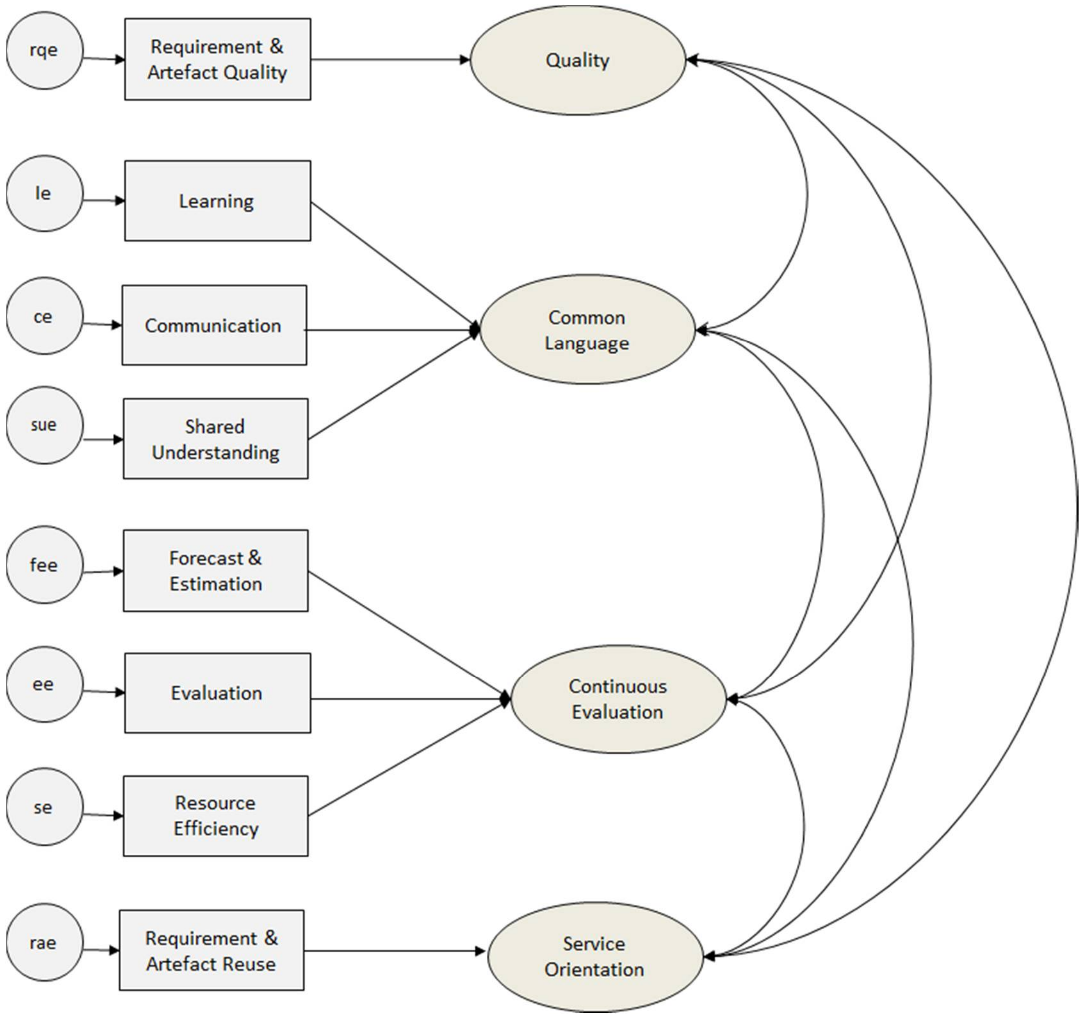

Figure 6.10: Study Structural Model

The estimated coefficients were assessed by calculating an overall coefficient of determination (R^2). This provided a relative measure of fit for each structural equation. The standardized estimation coefficients are summarized in Table 6.25. This model was implemented in LISREL 9.1 using the SIMPLIS approach.
The results of the LISREL analysis for this structure and the interpretation of the model with respect to the eight hypotheses of this study are presented in chapter seven.

6.5 Limitations and Summary

Some limitations of the data collection and analysis performed in this study have been highlighted. These include challenges with resource allocation to ensure that cross-pollination of procedures from the framework led portfolios to the ad hoc and third party portfolios.

Mitigating actions taken to address bias risk were introduction of the third party portfolio as control, concurrent execution of projects and balanced project portfolio composition. Factors such as IT stakeholder experience and skills may have had some impact on the results presented. It was challenging to find projects of similar complexity, duration and criticality to the business. These limitations notwithstanding, the data collection and analysis performed is considered sufficient enough to draw conclusions.

The data collection strategy, instrument design, descriptive statistical analysis and factor analysis performed have been discussed in this chapter. This forms the basis for chapter seven which covers interpretation of the results and theory synthesis.

Chapter 7 Interpretation of Results, Reflection and Theory Synthesis

7.1 Introduction

This chapter focuses on the interpretation of the results of the data analysis presented in chapter six. The objective is to reflect on the results and the significance of the findings with respect to stated objectives of the data collection and analysis stages of the study. The results are considered from the perspective of comparison of the four project portfolios and the levels of alignment attained in each and validation of the hypotheses in the study model. The assessment of the results in terms of the potential implications to research and practice are also discussed. The Alignment Forces Model (AFM) developed in the course of the research is introduced in this chapter. AFM is considered from the perspective of its usage in an iterative and agile SDLC setting.

A summary of the interpretation of the descriptive statistical analysis results is provided and reflection on validation of study model based on evidence from the results is undertaken. A comparative analysis of the REFINTO framework and support tool with alternative frameworks and tools is presented. Major findings of the study and its limitations of the study are highlighted.

7.2 Interpretation of Data Analysis Results

In this section, the interpretation of the analysis results, discussed to some extent in chapter six is expanded on. The second goal is to interpret the results of the REFINTO framework hypotheses validation.

7.2.1 Interpretation of Data Analysis Results by Category

The representation of the consolidated data for all four project portfolios and project stages is represented in a scatter diagram in Figure 7.1. It indicates that the ranking of alignment scores for the four portfolios in descending order is REFINTO and TOOL, REFINTO ONLY, THIRD PARTY, and NON-REFINTO.

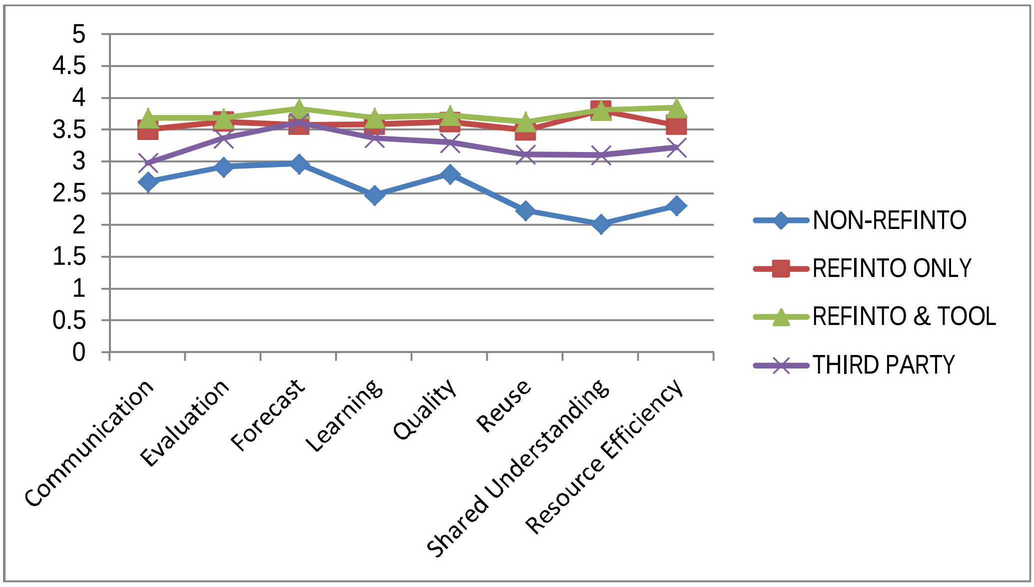

Figure 7.1: Scatter Diagram for Project Portfolios

The diagram also indicates better alignment for the REFINTO project portfolios on most of the eight factors. This implies that a structured and rigorous approach to project execution through the use of appropriate frameworks and tools can have positive impact on observed business-IT alignment maturity for a project portfolio. Detailed interpretation of the results for each project portfolio is presented in the following sub sections.

7.2.1.1 Interpretation of NON-REFINTO Portfolio Results

The NON-REFINTO project portfolio data represented in Figure 7.2 shows that shared understanding, learning, resource efficiency, and reuse scores were low.

Shared understanding and learning are important factors for engendering a common language and reducing the knowledge gaps between business and IT. Communication, the third factor for common language, was only marginally better.

Interestingly, the quality, evaluation, and forecast scores were better than the common language factors scores. This can probably be attributed to the experience, skills and domain knowledge of participants compensating for the limitations of using ad hoc requirements elicitation and project execution.

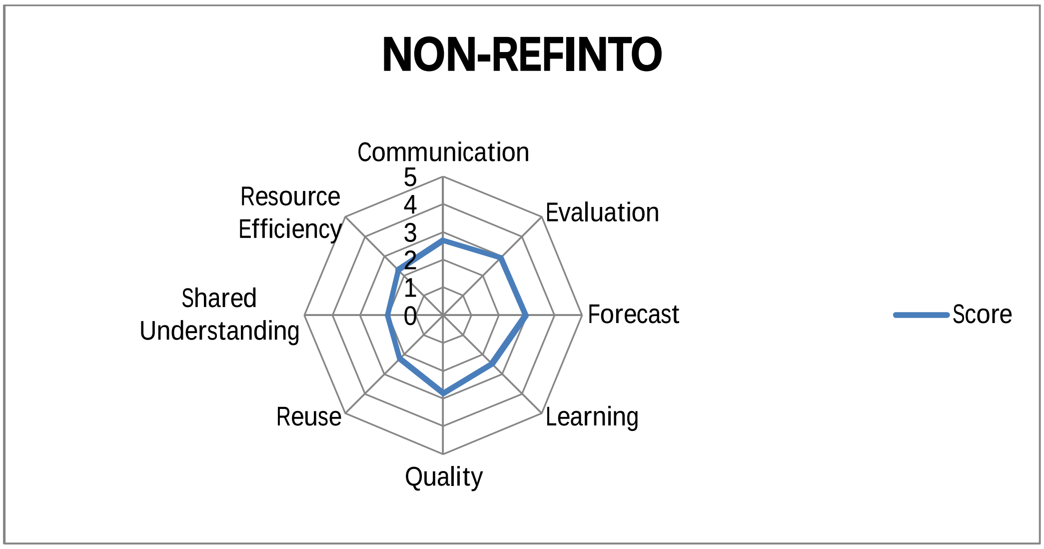

Figure 7.2: NON-REFINTO Portfolio Chart

7.2.1.2 Interpretation of REFINTO ONLY Portfolio Results

The REFINTO ONLY project portfolio data shown in Figure 7.3 indicates that shared understanding significantly improved in comparison to the NON-REFINTO portfolio. There were also considerable improvements in quality, learning and resource efficiency scores, although improvements in reuse and communication scores were less significant. This implies that as common language factors improve, alignment maturity also advances. The lower communication and reuse scores can be attributed to the manual processes used for requirements elicitation and refinement and requirements to artefact matching.

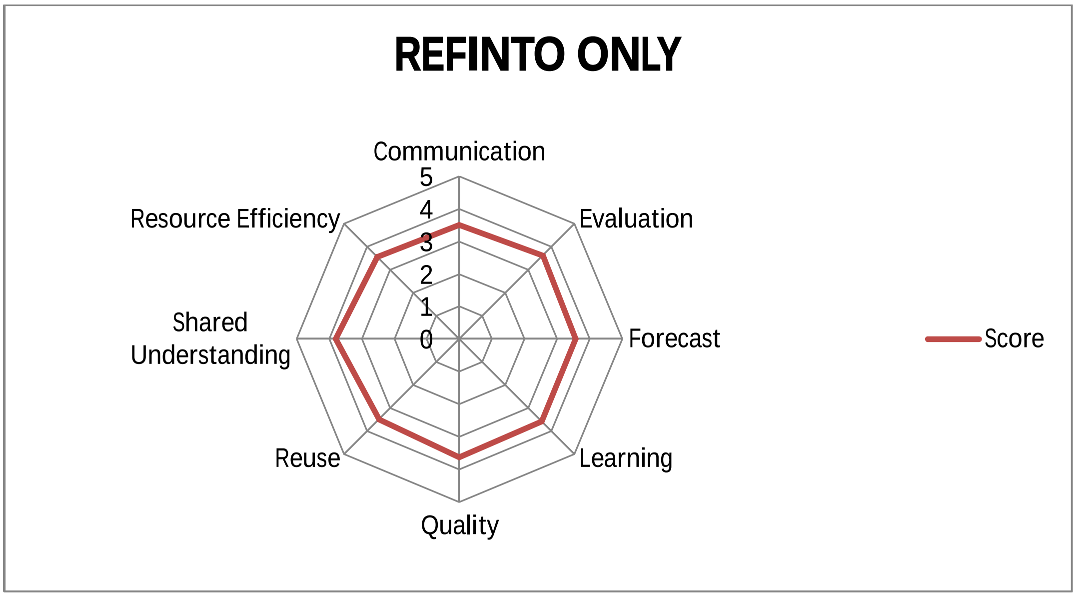

Figure 7.3: REFINTO ONLY Portfolio Chart

7.2.1.3 Interpretation of REFINTO & TOOL Portfolio Results

The REFINTO & TOOL project portfolio scores shown in Figure 7.4 indicate that communication and resource efficiency significantly improved in comparison to the REFINTO ONLY portfolio. There were noticeable improvements in learning, quality and forecast. Communication and reuse scores were expected to significantly improve due to tool support.

Interestingly, although reuse improved, it was to a lesser extent. This moderate improvement despite the use of tool support implies that reuse is difficult to achieve. This can also be attributed to the peculiarity of business needs that can affect the suitability of existing IT artefacts for satisfying new business needs. This is not surprising in rapidly evolving and innovative industries like the financial services. Another factor could be version incompatibility. This occurs when solutions are expected to be implemented in newer versions of APIs other than the versions used for implementing existing IT artefacts, regardless of downward compatibility. This can be due to architectural policies or security concerns, especially when open source APIs are used.

These results highlight the need for further refinement of the REFINTO framework and tool in future studies specifically focusing on improving reuse, communication, learning, and quality scores.

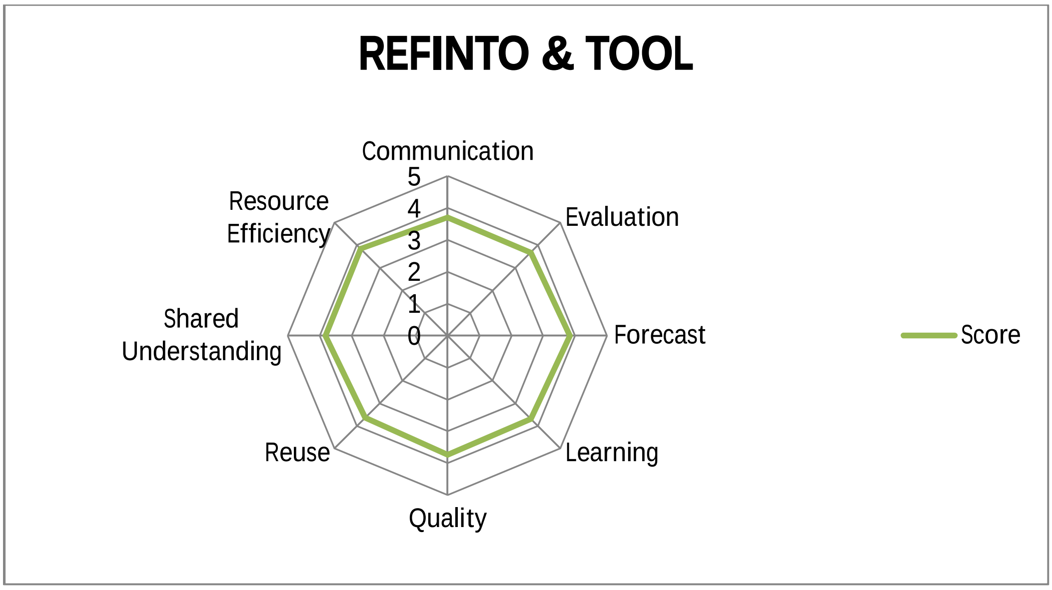

Figure 7.4: REFINTO & TOOL Portfolio Chart

7.2.1.4 Interpretation of THIRD PARTY Portfolio Results

The THIRD PARTY project portfolio data is shown in Figure 7.5.

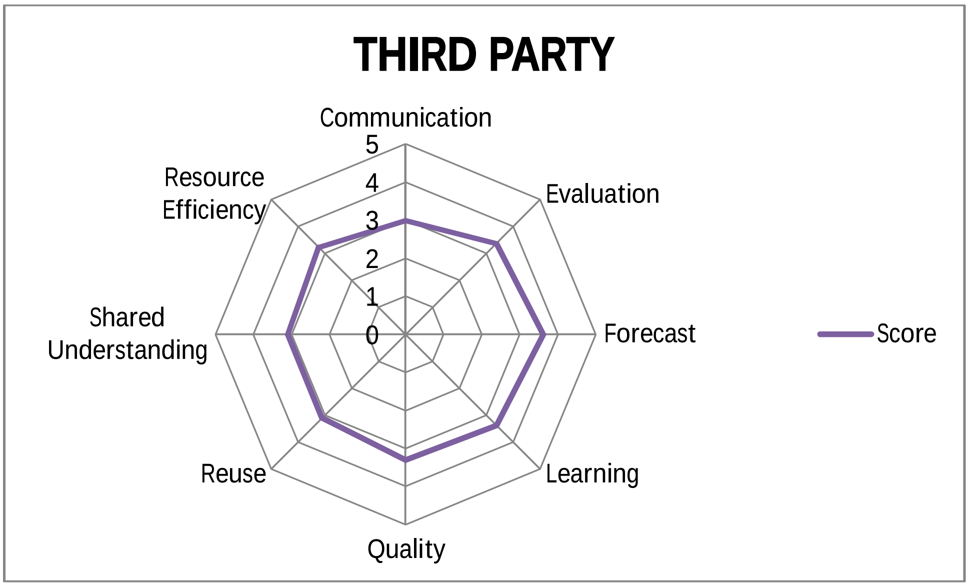

Figure 7.5: THIRD PARTY Portfolio Chart

It indicates better alignment scores in comparison to the NON-REFINTO portfolio but poorer compared to the REFINTO ONLY and REFINTO and TOOL portfolios. The scores observed on the eight factors measured indicate that roughly the same level (Level 3 Maturity - Established Focused Process) was attained.

The chart shows evenly spread scores which indicate that the THIRD PARTY processes produce a more balanced set of scores across the eight measures. In the context of comparing the THIRD PARTY portfolio to the REFINTO portfolios it presents an opportunity for improvements of the REFINTO framework and also an advantage that the REFINTO framework process has over the THIRD PARTY framework processes.

Firstly, it implies that refinements are needed in the REFINTO framework to improve on alignment factor scores that are lower. Secondly, it can also be interpreted that the REFINTO framework process can lead to improvements in key factors that have greater influence on overall alignment maturity.

7.2.2 Interpretation of Data Analysis Results by Stage

The interpretation of the data analysis results by project stage presented in chapter six to provide insight into the findings into the analysis of alignment maturity at the project stages is presented in this subsection.

7.2.2.1 Interpretation of Pre-Project Stage Results

The data for pre-project stage is represented in Figure 7.6. It indicates that the ranking of alignment scores for the four portfolios in descending order is REFINTO and TOOL, REFINTO ONLY, THIRD PARTY and NON-REFINTO.

The THIRD PARTY portfolio had significantly high scores on one factor (communication) which is almost at par with the REFINTO portfolios. This is attributable to the third party framework being communication-centric based on BABOK and PMBOK as highlighted in chapter six.

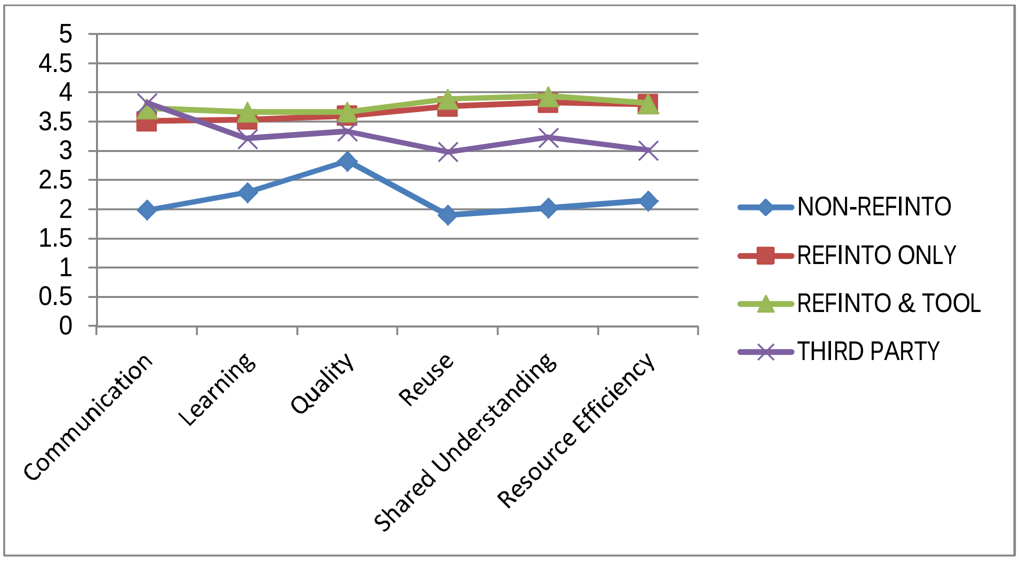

Figure 7.6: Scatter Diagram of Pre-Project Alignment for all Project Portfolios

7.2.2.2 Interpretation of Intra-Project Stage Results

The data for intra-project stage in Figure 7.7 indicates that the trend in ranking observed at the pre-project is correlated at this stage.

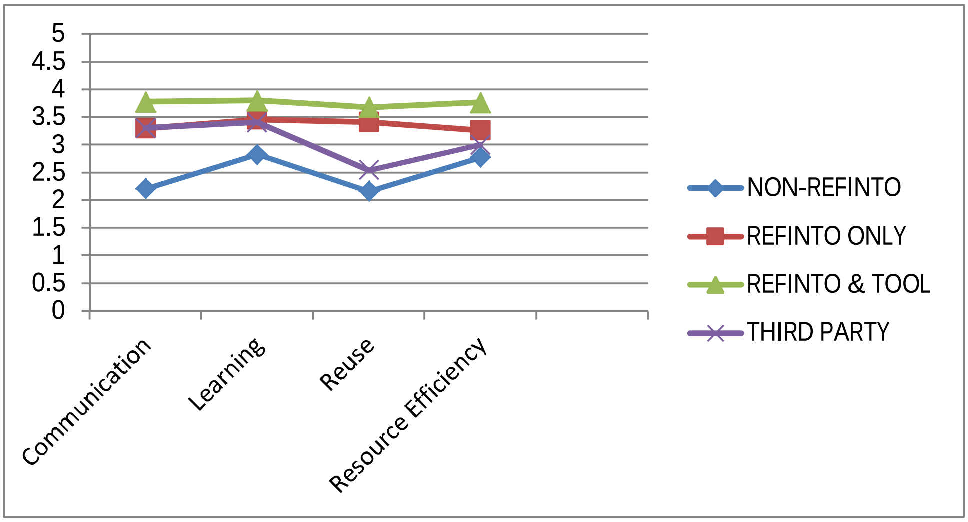

Figure 7.7: Scatter Diagram of Intra-Project Alignment for all Project Portfolios

The REFINTO & TOOL portfolio had better alignment scores at this stage. The differentiating factors at this stage are communication and reuse. This was expected because the REFINTO & TOOL portfolio had the benefit of tool support for requirement to artefact matching which aided reuse, more accurate project estimation, and better progress reporting.

The NON-REFINTO portfolio of projects had significantly lower scores for these two factors. The THIRD PARTY and REFINTO ONLY portfolios had similar scores for communication and learning. This further strengthens the argument that some form of structure and rigour in project conceptualization, planning, execution and governance can have significant impact on alignment maturity.

7.2.2.3 Interpretation of Post-Project Stage Results

The data for post-project stage represented in Figure 7.8 indicates correlation between alignment maturity ranking attained at the pre-project and intra-project stages.

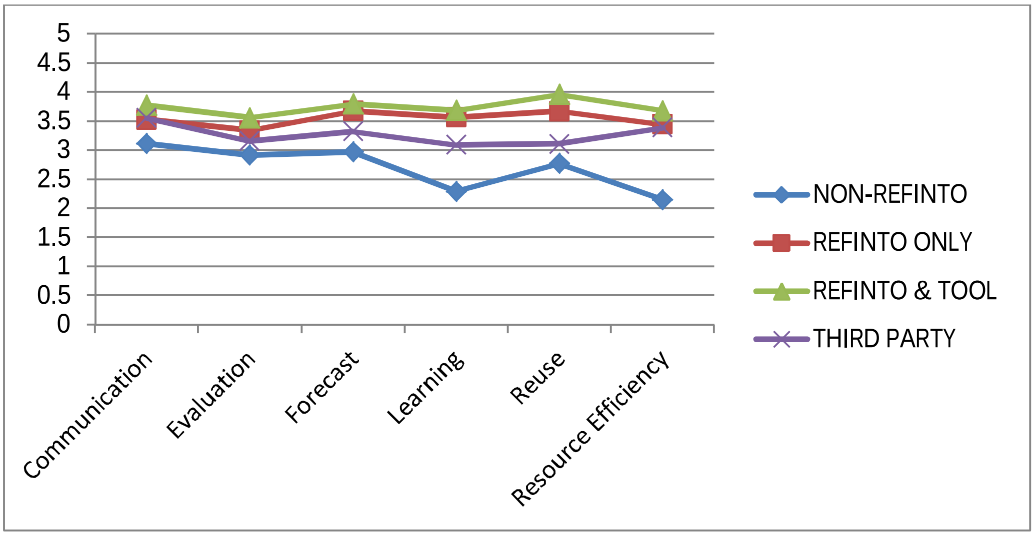

Figure 7.8: Scatter Diagram of Post-Project Alignment for all Project Portfolios

Learning and saving scores were low especially in the NON-REFINTO portfolio. This is significant in two ways. Firstly, it highlights the impact of learning on common language and by extension alignment. This implies that knowledge and language gaps are not being narrowed. Secondly, it highlights the influence of resource efficiency on continuous monitoring and consequently alignment. Interestingly, there seems to be a convergence of scores at improved/managed process ranking for resource efficiency in the REFINTO and TOOL, REFINTO ONLY, and THIRD PARTY portfolios.

The overall results also lend support to the argument that structured and rigorous approach to requirements elicitation, matching requirements to artefacts for reuse, project governance through continuous monitoring leads to better BIA.

7.3 Reflection on Hypotheses

The second objective of the data collection and analysis stage of the study was to validate the eight hypotheses in the study model. Descriptive statistical analysis and confirmatory factor analysis with factor structures hypothesized a priori was used for this purpose. This section focuses on the evidence obtained from the factor analysis to assess if findings supports and validates the eight hypotheses. The eight hypotheses form the endogenous factors of the model. The SEM model with both the path models and measurement models of the REFINTO framework is discussed in this section. The interpretation and statistical implication of the model estimation results are also discussed in this section.

In Table 7.1 and 7.2, one-way ANOVA performed on the consolidated data shows the test result is 80.04. The p-value for this statistics is $p<0.001$. This implies that there is evidence that there are differences in the means across the portfolios. The analysis shows that $F > F\ crit$ which implies that we can reject the null hypothesis. ANOVA does not however show where the differences are. As such a *t-Test* performed between the pairs, for illustration, REFINTO & TOOL versus NON-REFINTO. This is shown in Table 7.3.

Table 7.1: ANOVA Summary

Groups	Count	Sum	Average	Variance
NON-REFINTO	34	88.259	2.5959	0.1916
REFINTO ONLY	34	121.66	3.5782	0.0342
REFINTO & TOOL	34	126.531	3.7215	0.0339
THIRD-PARTY	34	109.881	3.2318	0.1674

Table 7.2: ANOVA Details

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	25.6440	3	8.5480	80.0370	0	2.6732
Within Groups	14.0977	132	0.1068			
Total	39.74162	135				

Table 7.3: t-Test: Two-Sample Assuming Equal Variances

	NON-REFINTO	REFINTO & TOOL
Mean	2.5959	3.7215
Variance	0.1916	0.0339
Observations	34	34
Pooled Variance	0.1128	
Hypothesized Mean Difference	0	
df	66	
t Stat	0	
P(T<=t) one-tail	0	
t Critical one-tail	1.6683	
P(T<=t) two-tail	0	
t Critical two-tail	1.997	

Since *t Stat* < -*t Critical* two-tall and t Stat is not greater then *t Critical* two tail, the null hypothesis cannot be rejected. The difference between the two is convincing enough to imply that alignment for the REFINTO & TOOL portfolio is significantly different from NON-REFINTO.

A two-step process was adopted for the SEM analysis. The measurement model was estimated separately and thereafter the estimation of the structural model was performed following Anderson and Gerbing (Anderson and Gerbing, 1984). LISREL (SIMPLIS approach) was used for the analysis. The SIMPLIS syntax used is depicted in Figure 7.9. An extract of the output from LISREL for the REFINTO framework model is shown in Figure 7.10.

The goodness-of-fit was done with root mean squared error of approximation (RMSEA) which at 0.047 shows a good fit (<0.05: good fit; <0.08: reasonable fit). The ratio chi-square/degrees of freedom is approximately 2 (df = 557, $\chi 2$/df = 2.04). A ratio in the range of 2–1 or 3–1 is indicative of an acceptable fit (Cote et al., 2001, Vieira, 2011). The goodness of fit index (GFI = 0.91), the adjusted goodness of fit index (AGFI = 0.90), the non-normed fit index (NNFI = 0.98), and the comparative fit index (CFI = 0.97), as well as the root mean square error of approximation (RMSEA = 0.047) indicate good fit (Vieira, 2011, Diamantopoulos and Siguaw, 2000, MacCallum et al., 1996). The fitted model is shown in Figure 7.11. In summary, the analysis indicates varying levels of support for the eight hypotheses. This is explored further in the next section.

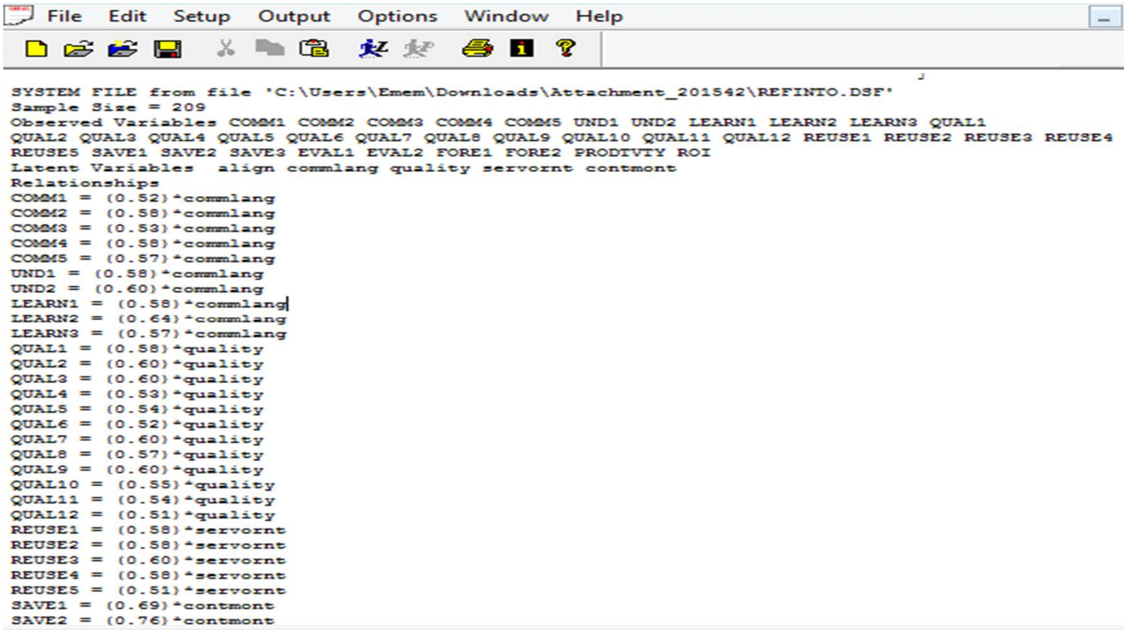

Figure 7.9: LISREL SIMPLIS Program for REFINTO Framework Model

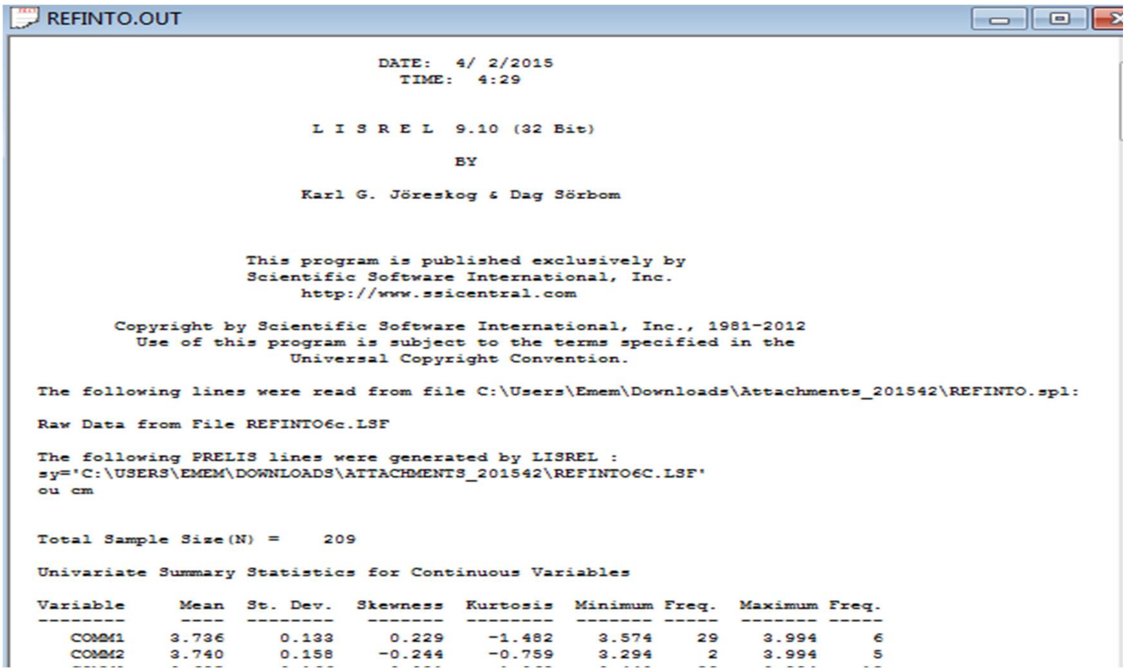

Figure 7.10: LISREL Output for REFINTO Framework Model

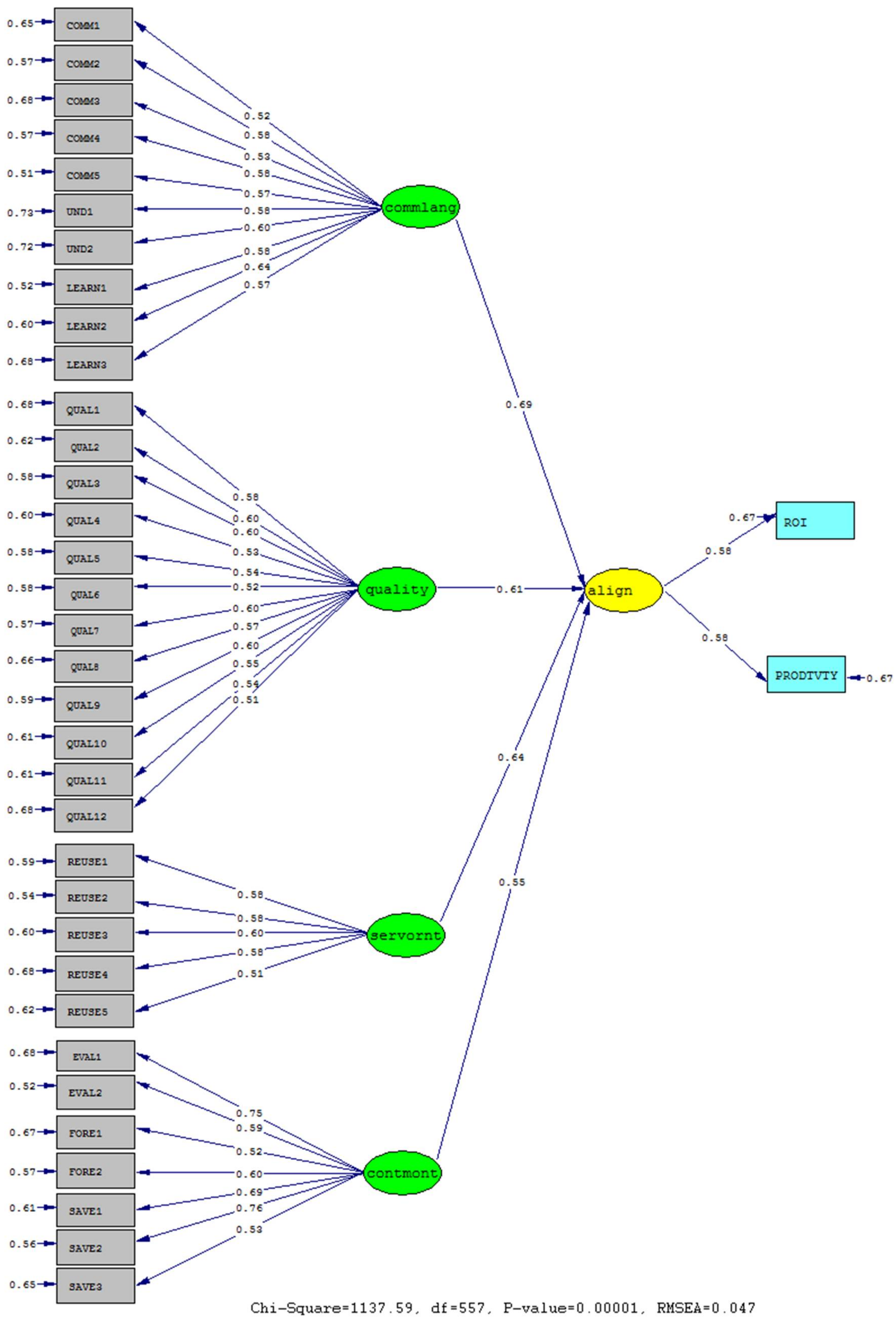

Figure 7.11: SEM Fitted Model in LISREL for REFINTO Framework

7.3.1 Quality Hypothesis

The quality hypothesis (H1) posits that the quality of business requirements has an influence on software product quality which subsequently positively impacts on business-IT alignment maturity at the tactical and operational levels.

The impact of the requirement and artefact quality on alignment maturity from the data analysis results was evaluated to give an indication of the validity of the quality (QUAL) hypothesis. The two factors in the H1 hypothesis are requirements quality (constructs QUAL1 to QUAL11) and artefact quality (QUAL12). The scores such as QUAL4 loading at 0.53, R^2 = 0.60 and correlation loading factor of 0.61 for common language indicates a high correlation between quality and alignment. Therefore, based on the results, H1 is supported.

7.3.2 Common Language Hypotheses

The common language hypotheses were shared understanding of domain knowledge (H2), communication (H3) and learning (H4). The support and validation of these hypotheses are reflected on in this sub section.

7.3.2.1 Shared Understanding/Domain Knowledge

The shared understanding hypothesis (H2) posits that shared knowledge and mutual understanding between business and IT has positive impact on business-IT alignment at tactical and operational levels.

The scores for shared understanding UND1 and UND2 had loading factors at 0.58 and 0.60 respectively, while R^2 are 0.73 and 0.72 respectively. This indicates a high correlation between shared understanding and alignment through common language (COMMLANG). Therefore, based on the results, H2 is supported.

7.3.2.2 Communication

The communication hypothesis (H3) posits that effective, timely, and frequent communication providing visibility of business process, impact of change, issues such as project estimates positively impacts with business-IT alignment maturity at tactical and operational levels.

The communication scores (COMM1-COMM5) such as COMM3 loading at 0.53 and R^2 = 0.68 indicates a high correlation between communication and alignment through common language (COMMLANG). Therefore, based on the results, H3 is supported.

7.3.2.3 Learning

The communication hypothesis (H4) posits that learning is positively impacts on business-IT alignment maturity at tactical and operational levels. The learning scores (LEARN1, LEARN2 and LEARN3) such as LEARN2 loading at 0.64 and R^2 = 0.60 indicates a high correlation between learning and alignment through common language (COMMLANG). Therefore, based on the results, H4 is supported.

7.3.3 Service Orientation Hypothesis

The service orientation hypothesis (H5) encompasses requirement and arterfact reuse. It posits that reuse of requirements, constraints, and other IT artefacts can lead to time and financial savings, learning and skills, and subsequently influence business-IT alignment at tactical and operational levels positively.

The two factors in the H5 hypothesis are requirements reuse (constructs REUSE1, REUSE2 and REUSE3) and artefact reuse (REUSE4 and REUSE5). The scores, such as REUSE4 loading at 0.58 and R^2 = 0.68 indicate high correlation between reuse and alignment through service orientation (SERVORNT) Therefore, from the results H5 is supported.

7.3.4 Continuous Monitoring Hypothesis

The continuous monitoring hypotheses encompass evaluation (H6), forecast and estimation (H7), and saving (H8). Support and validation of these hypotheses are reflected on in this subsection.

7.3.4.1 Evaluation

The evaluation hypothesis (H6) posits that continuous evaluation of processes and outcomes and codification of the same for future use positively impacts on business-IT alignment at tactical and operational levels.

The scores for evaluation EVAL1 and EVAL2, shows loading factors at 0.75 and 0.59 respectively, while R^2 are 0.68 and 0.52 respectively, indicating a high correlation between evaluation and alignment through continuous monitoring language (CONTMONT). Therefore, based on the results, H6 is supported.

7.3.4.2 Forecasting and Estimation

The forecast and estimation hypothesis (H7) posits that accurate forecasting and estimation leads to confidence and mutual trust between business and IT functions and is positively impacts business-IT alignment maturity at tactical and operational levels.

The scores for evaluation FORE1 and FORE2, shows loading factors at 0.52 and 0.60 respectively, while R^2 = 0.67 and R^2 = 0.57 respectively indicating a high correlation between forecast and estimation and alignment through continuous monitoring language (CONTMONT). Therefore, from the results H7 is supported.

7.3.4.3 Resource Efficiency

The resource efficiency hypothesis (H8) posits that cost and time savings is both a benefit from and an influence on business-IT alignment at the tactical and operational levels. The scores for resource efficiency SAVE1, SAVE2 and SAVE3 shows loading factors at 0.69, 0.76 and 0.53 respectively, while R^2 = 0.61, 0.56, 0.68 respectively indicating a high correlation between resource efficiency and alignment through continuous monitoring language (CONTMONT). Therefore, from the results H8 is supported.

7.4 Theory Synthesis from Data Analysis

On completion of the data analysis stage of the research, consideration was given to generalizing the knowledge and insights gained from the research in such a way that it has practical relevance to domains other than the financial services. The findings were considered as having significance to research and practice. As such, consideration was given to formulation of a theory from the research. Synthesis of theory on knowledge-based approach to Business-IT Alignment at tactical and operational levels that would contribute to the body of knowledge on business-IT alignment and be useful as a guide for practitioners in industry was considered.

The process of theory development in software engineering has a number of parts namely, constructs, propositions, explanations, and scope (Sjøberg et al., 2008). This guidance was followed to distil the study into one holistic model. This led to the proposal of a knowledge-based, requirements engineering-centric model, that maps software development and project management lifecycles to BIA processes for project execution at functional levels (tactical and operational). This model, the '*Alignment Forces Model*', is envisaged to be useful to academics and practitioners and is discussed in this section.

7.4.1 Alignment Forces Model (AFM)

Based on the empirical evaluation and validation of the eight hypotheses of the REFINTO framework and the subsequent positive results obtained, a model of the forces (factors or antecedents) influencing alignment at tactical and operational levels is proposed. The model is shown in Figure 7.12.

The constructs that make up the forces are simply the factors (antecedents) that influence business-IT alignment at tactical and operational level. These are from the hypotheses of the study model discussed in this chapter and in chapters five and empirically validated in this chapter and chapter six.

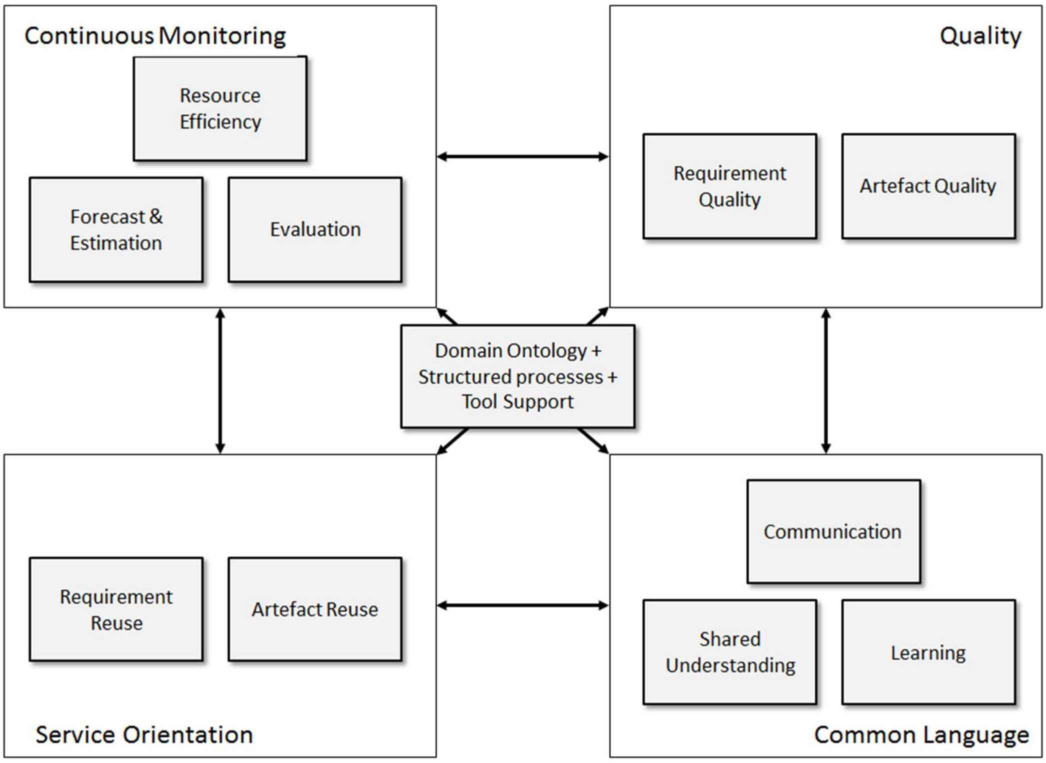

Figure 7.12: Alignment Forces Model

At the core of the model are its pre-requisites namely the domain ontology, structured processes, and optionally, a tool for semi-automated support for the processes. This facilitates structured and rigorous approach to project conceptualization, planning, execution and governance with business-IT alignment as a first class citizen of SDLC and PMLC as argued for in this thesis. Tool support is optional but necessary if optimal efficiency is desired as the data analysis results for the REFINTO & TOOL and REFINTO ONLY project portfolios indicate.

The four quadrants represent the categorizations of the hypotheses namely quality, common language, service orientation, and continuous monitoring which have been discussed in earlier chapters of the thesis. The arrows connecting the four quadrants illustrate the interaction and interplay between the factors in the model, resulting in a unified force to drive sustainable and optimal business-IT alignment. The model supports a process for project initiation, execution and governance for achieving business-IT alignment through knowledge-based structured, rigorous and agile processes with collaboration between business and IT stakeholders during the requirements engineering stage of the project and continuing throughout the SDLC. The model can be followed in a clock-wise direction starting from the first quadrant - quality quadrant focusing on achieving high requirements quality using the *elicit-refine-persist* triad.

The second quadrant (going clock-wise) is the common language quadrant focusing on shared understanding, learning, and effective communication. With common language in place the third quadrant (service orientation), focusing on using existing requirements and associated artefact, where relevant, using the *requirement-ontology-artefact* triad for reuse is initiated. At this stage, users of the model are empowered with sufficient information to forecast and estimate resources (in time and financial terms) required for implementing the requirements. The foundation for achieving higher levels of alignment between business and IT is therefore established.

The fourth quadrant (continuous monitoring), focusing on evaluation of the project outcomes and the resource efficiency attained from using the process is reached. The cycle ends in the quality quadrant with artefact quality assessment with expectation that artefact quality correlates with requirements quality. The cycle can be repeated as requirements change, thereby supporting agile and iterative development methodologies.

The Alignment Forces Model takes into consideration the interactive, iterative and evolutionary nature of agile software development practices which can be leveraged to attain business-IT alignment at operational and tactical levels of the organization, the main thesis of this study. AFM can be used complementarily with other software development methodologies.

7.4.2 Mapping AFM to SDLC

AFM can be mapped to the three REFINTO framework project stages as well as the SDLCs stages as presented in Table 7.4. It indicates how the AFM can be applied in practice. When maintenance is required or additional features are needed, the cycle is simply repeated in adherence to agile and iterative development methodology.

Table 7.4: Mapping AFM to SDLC and REFINTO Framework Project Stages

SDLC Stage	REFINTO Project Stage	AFM Quadrant
Requirements	Pre-Project	Common Language, Service Orientation
Design	Intra-Project	Common Language, Service Orientation
Implementation	Intra-Project	Common Language, Service Orientation
Verification	Post-Project	Continuous Monitoring, Quality
Maintenance	Repeat Cycle	Repeat Cycle

7.4.3 AFM – Implications to Researchers

AFM has potential to be useful to researchers as a model to base further research on business-IT alignment. The model provides a means of linking research in four distinct information systems and software engineering fields namely requirements engineering, SDLC, PMLC, and business-IT alignment into a holistic research endeavour. This can potentially facilitate maximizing return on investment in IT investments and fostering convergence of business and IT functions into coherent and coordinated processes at functional levels for execution of business strategy with particular attention on business-IT alignment.

7.4.4 AFM – Implications to Practitioners

AFM has potential benefits to practitioners seeking tangible and practical means of attaining sustainable business-IT alignment in a number of ways. Firstly it provides defined processes for conceptualization, planning, execution, and governance of projects with business-IT alignment as a priority. Secondly, AFM is well suited to iterative and agile software development methodologies widely adopted in industry in recent times. Thirdly, AFM can be used as a tool for organizational procedural and policy formulation that has business-IT alignment as the goal. Fourthly, it can be useful as a learning tool for IT stakeholders with potentials for improvements in employee productivity, organization performance and increased return on investment in IT.

7.5 Major Findings from Results Analysis

The main findings from the data analysis results are that the hypotheses and study model are supported. It also indicates support for arguments made for a knowledge-based requirements engineering framework towards attaining sustainable business-IT alignment maturity. Other key findings from the data analysis results are:

1. It can be deduced from the results that some form of structure and rigour in requirements engineering, software implementation, project execution, and governance can lead to improved alignment maturity. Therefore it can be argued that adopting an appropriate framework for these tasks is better than using ad hoc processes if sustainable business-IT alignment is to be attained.

2. It was observed that even with the use of the REFINTO framework and tool, improvement in reuse scores was not as significant as first envisaged. This supports the argument in extant literature that although reuse is desirable, it is challenging and difficult to achieve (Sherif and Vinze, 2003, Garlan et al., 1995, Morisio et al., 2002, Tracz, 1988). This notwithstanding, the gains of using knowledge-based requirements engineering framework guided processes did lead to improvements in reuse with positive impact on quality of requirements, time and cost reduction, learning about the domain, and making explicit the context in which requirements and associated artefacts were applicable in the domain. This invariably has positive impact on business-IT alignment.

3. The alignment scores for all eight factors for projects in the REFINTO portfolios and three project stages were below the Optimized Process mark

(4.5). This supports the argument that attaining optimal business-IT alignment maturity is difficult (Luftman, 2004, Jahnke, 2004). This notwithstanding, improvements in bridging the knowledge and language gaps between business and IT stakeholders, and a disciplined approach to project conceptualization, execution, and governance have been shown to have potentials for moving towards optimal business-IT alignment.

7.6 Limitations and Summary

In this chapter consideration has been given to the interpretation of data analysis results. Comparison of the four project portfolios were analysed further and the results assessed and reflected upon. Factor analysis performed to validate the framework hypotheses was interpreted. The alignment forces model, synthetized from insights gained from the data analysis was presented.

In a study of this magnitude, limitations and areas of weaknesses are inevitable. These limitations are the basis of the recommendations for further studies in chapter eight. The limitations are summarized here.

1. The size and scope of the data collection based on ten projects in each portfolio is still relatively small. It can be extended to further validate the results obtained and the claims made in this study. Since the data used were collected from real-life, business-critical projects over about four years, it would therefore require significantly longer time to realistically achieve.

2. The lack of full control in assigning the same set of participants in the study to a given project portfolio and keeping them on that category of projects for the duration of study was a limitation. It was possible that participants in the REFINTO portfolios when moved to projects using the THIRD PARTY or NON-REFINTO would have used the experience gained to positively impact on the scores of the THIRD PARTY and NON-REFINTO portfolios.

3. The REFINTO framework has been assessed with only one model. In some studies, a control path such as like organizational size (Charoensuk et al., 2014) is used.

Suggestions to extend the REFINTO study model by adding other factors are made in chapter eight.

Chapter 8 Contributions, Conclusions and Future Work

8.1 Introduction

This thesis started with the review of the current state of business-IT alignment research which revealed a number of gaps. Firstly, the focus solely on alignment at the strategic level has so far not delivered and seems unlikely to deliver sustainable business-IT alignment. Secondly, some proposed metrics and alignment measurement models targeted at the strategic level are difficult to operationalize. Thirdly, practical models that have business-IT alignment as first class citizen of software development and project management lifecycles is lacking. Fourthly, proposed antecedents to business-IT alignment are predominantly for strategic levels, leaving a gap for identifying antecedents to operational and tactical alignment levels. This study has attempted to contribute to knowledge by addressing these gaps by exploring a means of actualizing business-IT alignment in practice. A framework, support tool, rigorous, and structured processes applied to projects at operational and tactical levels of an organization have been proposed and validated.

Various arguments have been made and extant literature cited to support the approach proposed for actualizing the elusive business-IT conundrum. Firstly, it is argued that the interaction and collaboration between business and IT stakeholders during requirements engineering activities can be leveraged to bridge language and knowledge gaps between these stakeholders and thereby drive sustainable business-IT alignment. Conversely, ad hoc and poorly structured requirements elicitation, management, and eventual project execution are identified as factors that can exacerbate language and knowledge gaps between business and IT stakeholders, and become an inhibitor of business-IT alignment. Eight hypotheses, grouped into four categories are proposed as antecedents to operational and tactical alignment.

These antecedents are referred to as *pillars* of or *forces* influencing sustainable business-IT alignment.

These hypotheses, processes, metrics, and measurement models which are central to the REFINTO framework are empirically validated using a qualitative data collection instrument and quantitative statistical analyses. The REFINTO tool, its orthogonal application to REFINTO framework processes and potentials for application to third party processes have been highlighted. The framework is by no means a silver bullet to all business-IT alignment problems. Its limitations, weaknesses, and areas needing improvements are highlighted. Other frameworks and their areas of strength and advantages over the REFINTO framework are pointed out.

This chapter concludes the thesis. It highlights the contributions that have been made in the course of this study. The research questions, findings, contributions to knowledge, and understanding of business-IT alignment, suggestions for further study on the subject that researchers interested in extending the study are put forward. Finally, the implication of the study to research and practice is highlighted.

8.2 Finding on Research Questions

The summary of findings in the study with respect to each of the research questions that were formulated at the inception of the study and validated through data collection and analysis are summarized in this section.

Main Research Question (MRQ):

The main research question (MRQ) investigated the viability of an ontology-based requirements engineering framework and tool approach applied to agile/RAD applications projects and if this can cumulatively contribute to improved operational and tactical business and IT alignment. The data analysis results indicate support for the argument that the use of the REFINTO framework and support tool has positive influence on improving requirement and IT artefact, quality, shared understanding, communication, and learning. The results also support the argument that the framework positively contributes to narrowing language and knowledge gaps between business and IT stakeholders as evidenced in the alignment scores for portfolios using the framework and tool.

The correlation between the alignment scores for each project in the four portfolios and the overall scores for the portfolios indicate consistency with the processes followed. It can therefore be argued that this indicates some degree of confidence that operational and tactical alignment can be predicted with a fair chance of accuracy. This implies that over a period of time, alignment maturity improvements at the micro-foundational level can contribute to macro-alignment maturity improvements. This argument can be strengthened by extending this study to a larger scale, with larger project portfolios, larger number of participants, longer period of study, and in multiple organisations.

***Sub Research Question 1 (SRQ1)*:**

The objective of the first sub research question (SRQ1) was to assess the impact the choice of requirements elicitation methods, processes, structures, and management practices during project conceptualization, planning, and execution has on alignment maturity at operational and tactical levels. The question was to investigate if the proposed framework in MRQ would support knowledge-based requirements engineering, SDLC, PMLC processes and if this would contribute to business-IT alignment maturity improvements as opposed to following ad hoc processes.

The summary of findings indicates that the REFINTO framework and tool facilitates knowledge-based elicitation of high quality requirements using *the elicit-refine-persist* paradigm through collaborative requirement elicitation forums, requirement-to-artefact matching using the *requirement-artefact-ontology* paradigm, learning, and knowledge sharing between stakeholders, reuse of existing artefacts during implementation, communication of project estimates and progress throughout project implementation, and evaluation of the delivery process post implementation.

***Sub Research Question 2 (SRQ2)*:**

The objective of the second sub research question (SRQ2) was to investigated if the proposed framework would facilitate more structured and rigorous requirements engineering and management processes and practices than ad hoc and inconsistent processes and practices without encumbering the process with the demands of the waterfall development method such as delaying projects till formal requirements documents are signed off and inflexibility to requirements changes.

The results of the data analysis indicate support for the argument that structured and rigorous processes through the framework and tool have positive influence on business-IT alignment maturity outcomes. The iterative processes of the framework allow for requirements changes and for incremental development as requirements are validated and prioritized. The better alignments scores of the third party framework portfolio compared to those of the ad hoc portfolio strengthen this argument.

An aspect of these claims that requires further investigation is the tracking of the time spent on elicitation and if this impacts on the overall project times. However, reuse and production of high quality requirements through the structured and rigorous requirements engineering and management practices imply that whatever extra time is expended, it is made up for by lower incidence of reworked or scrapped IT artefacts. The results of the data analysis, does not show evidence that the structured and rigorous processes introduced had any negative impact on project execution times and alignment maturity. The time taken for stakeholders to learn the processes of the framework is compensated by increased productivity.

Sub Research Question 3 (SRQ3):

One of gaps observed in extant literature is the lack of metrics and maturity measurement models for operational and tactical alignment. SAMM and other maturity models based on it and the metrics used in these models are conceptualized to mainly measure alignment at the strategic level and are difficult to operationalize. This motivated the third sub research question (SRQ3), aimed at identifying metrics and proposal of maturity measurement models that are suitable for the operational and tactical levels. It was also an objective to apply the identified metrics and proposed maturity measurement models to gauging alignment maturity for individual projects and portfolio of projects.

In the study, metrics that can be operationalized are identified and used to gauge business-IT alignment maturity for three stages of projects (pre-project, intra-project, and post-project). A measurement model based on measures from SAMM, BSC, extant literature, and industry practice experience is proposed. The metrics and measurement model are validated through application to business projects in the financial services domain.

Sub Research Question 4 (SRQ4):

In extant literature, inhibitors and enablers of strategic business-IT alignment are identified. Identification of antecedents to operational and tactical business-IT alignment is largely unexplored in extant literature. The fourth sub research question (SRQ4) is aimed at addressing this gap. Eight antecedents to business-IT alignment at operational level and tactical levels, grouped into four categories, are put forward and are empirically validated using statistical analysis like structural equation modelling (SEM). These are quality (requirements and artefact quality), service orientation (requirements and artefact reuse), common language (communication, shared understanding, and learning), and continuous monitoring (evaluation, forecast and evaluation, resource efficiency).

Results of the empirical validation of the study model embodying these antecedents indicate strong support for common language and quality as antecedents to operational and tactical alignment. There is also support for continuous monitoring and service orientation. The results for service orientation (requirements and artefact reuse) indicate support for the argument that reuse can influence business-IT alignment maturity but also confirms arguments in extant literature that reuse is difficult to achieve in practice.

8.3 Research Contributions

The main contributions made in this study include the furtherance of knowledge and understanding of BIA by proposing a means to attaining and sustaining the elusive business-IT alignment concept in practice. The contributions also include the REFINTO framework, Alignment Forces Model (AFM), metrics, maturity measurement model for gauging business-IT alignment maturity at operational and tactical levels, and literature output in the form of published papers. The key contributions of this study are:

1. The identification and operationalization of metrics for antecedents to achieving business-IT alignment at tactical and operational levels validated through factor analysis of the study model and descriptive statistical analysis. The identified metrics have theoretical backing and are relevant to daily operations of organizations.

2. The measurement model for gauging business-IT alignment maturity on a per-project basis indicating alignment maturity, which cumulatively gauges the alignment maturity for individual projects and portfolio of projects, is proposed. This measurement model is envisaged to be useful to researchers and practitioners seeking objective and tangible means of gauging business-IT alignment maturity.

3. The REFINTO framework and support tool (Umoh and Sampaio, 2014) which facilitate knowledge-based processes using past requirements and other

artefacts for eliciting and refining current requirements, assessing the delivered artefacts, facilitating closer involvement, interaction and collaboration between business and IT stakeholders together is proposed. The framework and tool have been found to be useful for narrowing the knowledge and language gaps between business and IT stakeholders. Results of data analysis indicate that the argument that the framework can have a positive influence on business-IT alignment maturity has merits. The support tool can be used orthogonally with the REFINTO framework or third party frameworks to offer semi-automated functionality for requirements elicitation, refinement, persistence, and management using ontology-based requirement matching and reuse paradigms. This contributes to production of high quality requirements and subsequently high quality IT artefacts. Findings indicate support for the argument that this contributes to improved business-IT alignment maturity.

4. The proposed Alignment Forces Model, which unifies the framework, tool, and processes mapped to software development and project management lifecycles with business-IT alignment as first class citizen is proposed. The model is expected to be useful to practitioners and researchers.

5. The study contributions to literature in the form of peer-reviewed and published papers. These include an introduction of the framework (Umoh et al., 2011), a paper on measurement and evaluation of projects using the framework (Umoh et al., 2012), a paper on the demonstration of the support tool (Umoh and Sampaio, 2014), and a journal paper under review which presents the validation of antecedents to operational and tactical alignment and the alignment forces model. These papers reinforce the argument for a practical means to attaining sustainable business-IT alignment starting at the operational and tactical levels. An attempt has been made to address gaps in extant literature which predominantly base analysis and validation of study models on surveys not backed by evidence-based practical application of proposed models. This makes the findings of this study more relevant to practice.

8.4 Future Work

There are extensions and refinements that can be made to the study to further validate and strengthen the claims made in this study. The suggestions are summarized here:

1. Expanding Project Portfolio and Participants

   The size of the project portfolios can be increased from forty business-critical projects used in this study. This will lead to increased confidence in the framework, models and processes proposed. The duration of the projects used in the study ranged from a couple of weeks to a few months. The study can be extended to include multi-year projects to further test the scalability and efficacy of the framework and tool. This will also entail significantly increasing the number of participants in the study.

2. Segregation of Project Participants

   One of the highlighted limitations of the study relates to cases where project participants were allocated to projects in multiple portfolios due to business constraints. In a future study, restriction of participants to specific portfolios for the duration of the study is recommended. This will mitigate the risk of cross-pollination of ideas and practices that may have some impact on alignment scores.

3. Variation of Study Model

   Variations of the study model by introducing factors such as organisational size can be done. This will further validate the study model and provide further insights into other factors that may influence alignment maturity at tactical and operational levels.

4. Improving Lower Outlying Alignment Scores

   Analysis of alignment scores for the REFINTO project portfolios indicated that some scores were significantly improved while others were less so. It would be interesting to conduct further analysis on the implications of this and on how to improve alignment scores on the factors with the lower scores. It is envisaged that this would have a positive impact on overall business-IT alignment maturity.

5. Maximizing Alignment Scores
   Analysis of alignment scores for all four project portfolios and at all three stages showed that there were no factors that met the optimized process mark (4.500 – 5.000). It would be interesting to explore further adjustments to the framework and support tool that may lead to improvements in alignment maturity to meet the optimized process mark.

Despite the need for further refinement of the REFINTO framework and support tool highlighted in this chapter, the framework and tool in their current form should be beneficial to organizations seeking to achieve improved business-IT alignment maturity in various ways. Firstly, they can be used as a template and platform for improved collaboration between stakeholders from the business and IT stakeholders throughout the software development and project lifecycle. Secondly, they facilitate learning, sharing domain knowledge, and communication throughout the development and project management lifecycles. Thirdly, they provide guidance and facilitate semi-automated and knowledge-driven reuse of artefacts and requirements to facilitate higher quality requirements and artefacts. Fourthly, they embody metrics and a measurement model that can be applied to objectively and tangibly gauge alignment maturity.

8.5 Implications for Research and Practice

This study has implications for research and practice towards realization of the goal of attaining sustainable business-IT alignment maturity. The REFINTO framework and support tool were conceptualized to provide a means of actualizing this goal. The study contributes to addressing criticism that business-IT alignment is vague, difficult to actualize, measure, and sustain (Umoh et al., 2011). The framework and tool facilitate reuse of historic requirements and artefacts thereby potentially leading to cost and time savings, higher productivity, and improved return on investment (ROI) in IT. It also bridges the knowledge and language gaps between business and IT stakeholders by facilitating a collaborative approach to requirements engineering, software development project execution, management, and evaluation. This provides an approach to ensure better visibility of project governance process and project status to all stakeholders.

The constructs and metrics applied in the framework can be easily replicated irrespective of the domain of practice. Practitioners will find the data and evaluation features of DEMM useful in day-to-day monitoring and evaluation of projects at all stages. Researchers will gain insight on directions to focus their efforts at finding practical means of actualizing sustainable business-IT alignment. This can contribute positively to enabling the uptake of business-IT alignment approaches with clear mechanisms for measuring effectiveness.

References

.
1998. IEEE Recommended Practice for Software Requirements Specifications. *IEEE Standard 830-1998*, 1-40.

2003. Board Briefing on IT Governance.

2014. Requirements Management- A Core Competency for Project and Program Success, Pulse of the Profession Project Management Institute.

ABDI, M. & DOMINIC, P. Strategic IT Alignment with Business Strategy: Service Oriented Architecture Approach. 2010. 1473-1478.

AGARWAL, B., TAYAL, S. & GUPTA, M. 2010. *Software Engineering & Testing: An Introduction,* Sudbury, Massachusetts, Jones and Barlett Publishers.

AHUJA, S. 2012. Strategic Alignment Maturity Model (SAMM) in a Cascading Balanced Scorecard (BSC) Environment: Utilization and Challenges. *CAISE 2012 Workshops.*Berlin Heidelberg: Springer-Verlag.

AIER, S. A. W., R 2009. Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. *WIRTSCHAFTSINFORMATIK, Business & Information Systems Engineering,* 2**,** 150-163.

AKERKAR, R. & SAJJA, P. 2010. *Knowledge-Based Systems,* London, Jones and Bartlett Publishers.

AL-HROUB, Y., KOSSMANN, M. & ODEH, M. Developing an Ontology-driven Requirements Analysis Tool (OntoRAT): A use-case-driven approach. Second International Conference on the Applications of Digital Information and Web Technologies, 2009. IEEE Press, 130-138.

AL BALUSHI, T. 2008. ELICITO: An Ontology-Based Quality Driven Requirements Engineering Framework and Tool. Manchester: University of Manchester.

AL BALUSHI, T. H., SAMPAIO, P. R. F. & LOUCOPOULOS, P. 2013. Eliciting and prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool. *Expert Systems,* 30**,** 129-151.

ALANA, E. & RODRIGUEZ, A. 2007. Domain Engineering Methodology. *CORDET* [Online]. Available: http://www.pnp-software.com/cordet/download/pdf/GMV-CORDET-RP-001_Iss1.pdf.

ALAVI, M. & LEIDNER, D. 1999. Knowledge management systems: issues, challenges, and benefits. *Communications of the AIS,* 1**,** 1-37.

ALDEA, A., BAÑARES-ALCÁNTARA, R., BOCIO, J., GRAMAJO, J., ISERN, D., KOKOSSIS, A., JIMÉNEZ, L., MORENO, A. & RIAÑO, D. 2003. An Ontology-Based Knowledge Management Platform. *IIWeb***,** 177-182.

ALMAJALI, D. & DAHALIN, Z. 2011. Factors Influencing IT-Business Strategic Alignment and Sustainable Competitive Advantage: A Structural Equation Modelling Approach. *Communication of the IBIMA***,** 1-12.

ANDERSON, J. C. & GERBING, D. W. 1984. The effect of sampling error on convergence, improper solutions, and Goodness-of-Fit indices for maximum likelihood confirmatory factor analysis. *Psychometricka,* 49**,** 155-173.

ANTONIOU, G., VAN HARMELEN, F., PLANT, R. & VANTHIENEN, J. 1998. Verification and Validation of Knowledge-Based Systems: Report on Two 1997 Events. *AI Magazine,* 19**,** 123-126.

ARBUCKLE, J. L. 1997. *AMOS User's Guide,* Chicago, Small-Waters.

ARIYACHANDRA, T. & FROLICK, M. 2008. Critical Success Factors in Business Performance Management - Striving for Success, Information Systems Management. *Information Systems Management,* 25**,** 113-120.

ARNDT, H. & KLEIN, R. 2002. An Ontology-Based Collaborative Framework for Decision Support in Engineering. *Digital Enterprise Challenges***,** 369-381.

ARPÍREZ, J. C., CORCHO, O., FERNÁNDEZ-LÓPEZ, M. & GÓMEZ-PÉREZ, A. WebODE: a scalable workbench for ontological engineering. 1st international conference on Knowledge capture, 2001. ACM, 6-13.

ARPÍREZ, J. C., CORCHO, O., FERNÁNDEZ-LÓPEZ, M. & GÓMEZ-PÉREZ, A. 2003. WebODE in a nutshell. *AI magazine,* 24**,** 37.

ASSAWAMEKIN, N., SUNETNANTA, T. & PLUEMPITIWIRIYAWEJ, C. 2010. Ontology-based multiperspective requirements traceability framework. *Knowledge and information systems,* 25**,** 493-522.

AURUM, A. & WOHLIN, C. 2005. Requirements Engineering: Setting the Context. *Engineering and Managing Software Requirements.*

AVERSANO, L., GRASSO, C. & TORTORELLA, M. 2012. A Literature Review of Business/IT Alignment Strategies. *Procedia Technology,* 5**,** 462-474.

AVISON, D., JONES, J. & POWELL, W., D. 2004. Using and validating the strategic alignment model. *Journal of Strategic Information Systems,* 13**,** 223-246.

BABAR, A., COX, K., BLEISTEIN, S. & VERNER, J. Towards evolution of Strategic IT requirements. ACM Symposium on Applied Computing, 2007 Seoul, Korea. ACM, 1220-1227.

BABKIN, E. & POTAPOVA, E. 2010. Towards ontology-based methodology for requirements formalization. *Perspectives in Business Informatics Research***,** 73-85.

BALHARETH, H. B., LIU, K. & MANWANI, S. Achieving business and IT alignment from organisational learning perspectives. International Conference on Business Information Systems, 2012 Paris, France. World Academy of Science, Engineering and Technology, 937-940.

BANARJEE, J. & AZIZ, S. 2007. Enterprise Architecture (EA), Service Oriented Architecture (SOA), Solution Architecture (SOLA): The Missing Link? *SETLabs Briefings,* 5**,** 69-80.

BARNEY, J., WRIGHT, M. & & KETCHEN, D. J. 2001. The resource-based view of the firm: Ten years after 1991. *Journal of management,* 27**,** 625-641.

BASILI, V. The Experimental Paradigm in Software Engineering. International Workshop on Experimental Software Engineering Issues: Critical Assessment and Future Directions, 1993 London, UK. Springer-Verlag, 3-12.

BATARSEH, F. & GONZALEZ, A. 2013. *Validation of knowledge-based systems: a reassessment of the field,* Dordrecht, Springer Science Business Media.

BEATTY, J., FERRARRI, R., VIJAYAN, B. & GODUGULA, S. 2011. Seilevel's Evaluations of Requirements Management Tools: Summaries and Scores.

Seilevel [Online]. Available: http://www.seilevel.com/wp-ontent/uploads/RequirementsManagementToolWP_2.pdf.

BEIMBORN, D., SCHLOSSER, F. & WEITZEL, T. Proposing a Theoretical Model for IT Governance and IT Business Alignment. 42nd International Conference on Service Systems, 2009 Hawaii, USA. IEEE Press, 1-11.

BENBASAT, I., GOLDSTEIN, D. & MEAD, M. 1987. The Case Research Strategy in Studies of Information Systems. *MIS Quarterly,* 2**,** 369-386.

BENBYA, H. & MCKELVEY, B. 2006. Using coevolutionary and complexity theories to improve IS alignment: a multi-level approach. *Journal of Information Technology,* 21**,** 284-298.

BENTLER, P. M. 1995. *EQS Structural Equations Programs Manuals,* Encino, CA, Multivariate Software Inc.

BERGERON, F., RAYMOND, L. & RIVARD, S. 2004. Ideal patterns of strategic alignment and business performance. *Information & Management,* 41**,** 1003–1020.

BHAT, J. M., POOLOTH, K., MOORTHY, M., SINDHGATTA, R. & THONSE, S. 2007. Use of ontology for automating knowledge intensive business processes. *Ontologies***,** 435-459.

BHATTACHERJEE, A. 2012. *Social science research: principles, methods, and practices 2ed*.

BLEISTEIN, S., COX, K. & VERNER, J. Strategic alignment in requirements analysis for organizational IT: An integrated approach. 2005 ACM Symposium on Applied Computing, 2005 Santa Fe, New Mexico, USA. ACM, 1300-1307.

BLEISTEIN, S. J., AURUM, A., COX, K. & RAY, P. K. 2004. Strategy-Oriented Alignment in Requirements Engineering: Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach. *Journal of Research & Practice in Information Technology,* 36**,** 259-276.

BLEISTEIN, S. J., COX, K., VERNER, J. & PHALP, K. T. 2006. B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process. *Information and software technology***,** 846-868.

BLESSING, L. & CHAKRABARTI, A. 2009. *DRM, a Design Research Methodology,* Dordrecht Heidelberg, Springer.

BÖCKLE, G., POHL, K. & VAN DER LINDEN, F. 2005. A Framework for Software Product Line Engineering. *Software Product Line Engineering Foundations, Principles and Techniques.* Springer.

BOEHM, B. 1998. A Spiral Model of Software Development and Enhancement. *IEEE Computer,* 21**,** 61-72.

BON, J. V. & HOVING, W. 2007. Strategic Alignment Model Enhanced. *ISM Portal* [Online]. Available: http://www.ismportal.nl/nl/system/files/SAME-20071119.pdf [Accessed July 30].

BORCH, O. & HARTVIGSE, G. 1991. Knowledge-based systems for strategic market planning in small firms. *Decision Support Systems,* 7**,** 145–157.

BRACHMAN, R. & LEVESQUE, H. 2004. *Knowledge Representation and Reasoning,* Elsevier.

BRADLEY, R. V., PRATT, R., BYRD, T. A., OUTLAY, C. N. & WYNN, J. D. E. 2012. Enterprise Architecture, IT Effectiveness and the Mediating Role of IT Alignment in US Hopsitals. *Information Systems Journal,* 22**,** 97-127.

BREITMAN, K. K. & SAMPAIO DO PRADO LEITE, J. C. Ontology as a requirements engineering product. 11th IEEE International Requirements Engineering Conference, 2003. IEEE Press, 309-319.

BROY, M. 2013. Domain Modelling and Domain Engineering: Key Tasks in Requirements. *Perspectives on the Future of Software Engineering.* Berlin Heidelberg: Springer-Verlag.

BRYANT, B., GRAY, J. & MERNIK, M. Domain-Specific Software Engineering. FoSER 2010, 2010 Santa Fe, New Mexico, USA. 65-68.

BYRD, T. A., LEWIS, B. R. & BRYAN, R. W. 2006. The leveraging influence of strategic alignment on IT investment: An empirical examinations. *Information & Management,* 43**,** 308–321.

CAMPBELL, B., KAY, R. & AVISON, D. 2005. Strategic alignment: a practitioner's perspective. *Journal of Enterprise Information Management,* 18**,** 653-664.

CAPLINSKAS, A., LUPEIKIENE, A. & VASILECAS, O. 2003. The Role of Ontologies in Reusing Domain and Enterprise Engineering Assets. *Informatica,* 14**,** 455–470.

CARLSON, N. & LAPLANTE, P. 2014. The NASA automated requirements measurement tool: a reconstruction. *Innovations in Systems and Software Engineering,* 10**,** 77-91.

CARRILLO DE GEA, J., NICOLÁS, J., FERNÁNDEZ ALEMAN, J., TOVAL, A., EBERT, C. & VIZCAÍNO, A. 2011. Requirements Engineering Tools. *IEEE Software,* 28**,** 86-91.

CASTAÑEDA, V., BALLEJOS, L., CALIUSCO, M. L. & GALLI, M. R. 2014. A semantic web enabled approach to reuse functional models in web engineering. *Automated Software Engineering,* 22**,** 241-288.

CASTRO, J., KOLP, M. & MYLOPOULOS, J. 2002. Towards requirements-driven information systems engineering: the Tropos project. *Information systems,* 27**,** 365-389.

CHAN, Y. 2002. Why haven't we mastered alignment? The importance of the informal organization structure. *MIS Quarterly Executive,* 2**,** 97-112.

CHAN, Y. & REICH, B. 2007. State of the art IT alignment: what have we learned? *Journal of Information Technology,* 22**,** 297-315.

CHAN, Y., SABHERWAL, R. & THATCHER, J. 2006. Antecedents and Outcomes of Strategic IS Alignment: An Empirical Investigation. *IEEE Transactions on Engineering Management,* 53**,** 27-47.

CHANDRA, C. & TUMANYAN, A. Ontology driven knowledge design and development for supply chain management. 13th Annual Industrial Engineering Research Conference, 2004 Houston, TX. IERC, 1-6.

CHANDRASEKARAN, B. & SWARTOUT, W. 1991. Explanations in knowledge systems: the role of explicit representation of design knowledge. *IEEE Intelligent Systems,* 6**,** 47-49.

CHAROENSUK, S., WONGSURAWAT, W. & KHANG, D. 2014. Business-IT Alignment: A practical research approach. *The Journal of High Technology Management Research,* 25**,** 132-147.

CHEBROLU, S. & NESS, L. 2013. How Does Alignment of Business and IT Strategies Impact Aspects of IT Effectiveness? *International Journal of Applied Management and Technology,* 12**,** 1-15.

CHEMUTURI, M. 2013. *Requirements Engineering and Management for Software Development Projects,* London, UK, Springer.

CHEN, H. Towards service engineering: service orientation and business-IT alignment. 41st Annual International Conference on System Sciences, 2008 Hawaii, USA. IEEE Press, 114-124.

CHEN, L. 2010. Business–IT alignment maturity of companies in China. *Information & Management,* 47**,** 9-16.

CHEN, R., SUN, C., HELMS, M. & JIH, W. 2008. Aligning information technology and business strategy with a dynamic capabilities perspective: A longitudinal study of a Taiwanese Semiconductor Company. *International Journal of Information Management,* 28**,** 366-378.

CHEN, W. & HIRSCHHEIM, R. 2004. A paradigmatic and methodological examination of information systems research from 1991 to 2001. *Journal of Information Systems,* 14**,** 197–235.

CHIKH, A., ABULAISH, M., NABI, S. I. & ALGHATHBAR, K. 2011. An ontology based information security requirements engineering framework. *Secure and Trust Computing, Data Management and Applications***,** 139-146.

CHONG, E. E., NAZIM, A. & AHMAD, S. B. 2014. A Comparison between Individual Confirmatory Factor Analysis and Pooled confirmatory Factor Analysis: An Analysis of Library Service Quality, a Case Study at a Public University in Terengganu. *International Journal of Engineering Science and Innovative Technology,* 3**,** 110-116.

CIBORRA, C. 1993. De Profoundis? Deconstructing the Concept of Strategic Alignment. *Scandinavian Journal of Information Systems,* 9**,** 67-82.

CIBORRA, C. 2000. *From Control to Drift: The Dynamics of Corporate Information Infrastructures,* New York, Oxford University Press.

CLARK, T. & BARN, B. 2013. Domain Engineering for Software Tools, in Domain Engineering. *Domain Engineering.* Springer-Verlag Berlin Heidelberg.

COLLIS, J. & HUSSEY, R. 2009. *Business Research: A Practical Guide for Undergraduate and Postgraduate Students 3ed,* New York, Palgrave MacMillan.

COLTMAN, T. R., TALLON, P. P., SHARMA, R. & QUEIROZ, M. 2015. Strategic IT alignment: twenty-five years on. *Journal of Information Technology,* 30**,** 91-100.

COTE, J., NETEMEYER, R. & BENTLER, P. 2001. Structural equation modeling—improving model fit by correlating errors. *Journal of Consumer Psychology,* 10**,** 87–88.

COUGHLAN, J., LYCETT, M. & MACREDIE, R. 2005. Understanding the business-IT relationship. *International Journal of Information Management,* 25**,** 303-319.

COUTO, R., RIBEIRO, A. N. & CAMPOS, J. C. 2014. Application of Ontologies in Identifying Requirements Patterns in Use Case. *arXiv preprint arXiv:1404.0850***,** 62-76.

CRAGG, P., KING, M. & HUSSIN, H. 2002. IT alignment and firm performance in small manufacturing firms. *Journal of Strategic Information Systems,* 11**,** 109–132.
CROISIER, S. 2012. The Rise of Semantic-Aware Applications. *Semantic Technologies in Content Management Systems***,** 23-33.
CUMPS, B., VIAENE, S. & DEDENE, G. 2006a. Managing for better business-IT alignment. *IT Professional,* 8**,** 17-24.
CUMPS, B., VIAENE, S., DEDENE, G. & VANDENBULCKE, J. An Empirical Study on Business/ICT Alignment in European Organisations. 39th Annual International Conference on System Sciences, 2006b Hawaii, USA. IEEE Press, 195.
CZARNECKI, K. & EISENECKER, U. 2000. *Generative Programming - Methods, Tools, and Applications,* Boston, MA, Addison-Wesley.
DAG, J. & GERVASI, V. 2005. Managing Large Repositories of Natural Language. *Engineering and Managing Software Requirements.* Berlin Heidelberg: Springer-Verlag.
DARDENNE, A., VAN LAMSWEERDE, A. & FICKAS, S. 1993. Goal-directed requirements acquisition. *Science of computer programming,* 20**,** 3-50.
DECKER, B., RECH, J., RAS, E., KLEIN, B. & HOECHT, C. Selforganized Reuse of Software Engineering Knowledge supported by Semantic Wikis. Workshop on Semantic Web Enabled Software Engineering, 2005.
DEHLINGER, J. & LUTZ, R. R. Supporting requirements reuse in multi-agent system product line design and evolution. IEEE International Conference on Software Maintenance, 2008. 207-216.
DELISI, P. 2005. IT Alignment Revisited: Why we're no closer to IT alignment today than we were 20 years ago. *CIO Magazine*.
DEVEDZIC, V. 2004. Education and the Semantic Web. *International Journal of Artificial Intelligence in Education,* 14**,** 39-65.
DEVINNEY, T. M. 2013. Is Microfoundational Thinking Critical to Management Thought and Practice? *Academy of Management Perspectives,* 27**,** 81-84.
DIAMANTOPOULOS, A. & SIGUAW, J. 2000. *Introducing LISREL,* London, SAGE.
DJELOUAH, R., DUVAL, B. & LOISEAU, S. 2002. Validation and Reparation of Knowledge Bases. 312-320.
DOBSON, G. & SAWYER, P. Revisiting ontology-based requirements engineering in the age of the semantic web. International Seminar on Dependable Requirements Engineering of Computerised Systems at NPPs, 2006.
DODANI, M. 2007. Aligning IT to business through architecture. *Journal of Object Technology,* 7**,** 1-8.
DOHERTY, N. F., ASHURST, C. & PEPPARD, J. 2012. Factors affecting the successful realisation of benefits from systems development projects: findings from three case studies. *Journal of Information Technology,* 27**,** 1-16.
DOLL, W. J. & TORKZADEH, G. 1988. The Measurement of End-User Computing Satisfaction. *MIS Quarterly,* 12**,** 259-274.

DVIR, D., RAZ, T. & SHENHAR, A. 2003. An empirical analysis of the relationship between project planning and project success. *International Journal of Project Management,* 21**,** 89–95.

EASTERBROOK, S. M., SINGER, J., STOREY, M. A. & DAMIAN, D. 2008. Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Engineering.* Springer.

EDGE, M. 2010. A Policy Driven Approach to Proactive Fraud Management in Financial Data Streams. Manchester: University of Manchester.

EDWARDS, M. L., FLANZER, M., TERRY, M. & LANDA, J. RECAP: a requirements elicitation, capture and analysis process prototype tool for large complex systems. Engineering of complex computer systems, 1995. IEEE Press, 278-281.

EISENHARDT, K. M. & MARTIN, J. A. 2000. Dynamic capabilities: what are they? *Strategic Management Journal,* 21**,** 1105–1121.

ENDRES, A. & ROMBACH, D. 2003. *A Handbook of Software and Systems Engineering* **-** *Empirical Observations, Laws and Theories,* Pearson Education.

ERIKSSON, H. 2003. Using JessTab to Integrate Protégé and Jess. *IEEE Intelligent Systems,* 18**,** 43-50.

ESMAILI, H., GARDESH, H. & SIKARI, S. Strategic alignment: ITIL perspective. 2nd International Conference on Computer Technology and Development, 2010. 550-555.

FAIRLEY, R. 2009. *Managing and Leading Software Projects,* Hoboken, New Jersey, IEEE Computer Society.

FALBO, R. D. A., GUIZZARDI, G. & DUARTE, K. C. 2002. An ontological approach to domain engineering. *14th international conference on Software engineering and knowledge engineering***,** 351-358.

FARFELEDER, S., MOSER, T., KRALL, A., STÅLHANE, T., OMORONYIA, I. & ZOJER, H. 2011. Ontology-driven guidance for requirements elicitation. *The Semanic Web: Research and Applications***,** 212-226.

FINKELSTEIN, A. 1988. Re-use of formatted requirements specifications. *Software Engineering Journal,* 3**,** 186-197.

FIRESMITH, D. 2007. Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them. *Journal of Object Technology,* 6**,** 17-33.

FRANCH, X., PALOMARES, C., QUER, C., RENAULT, S. & DE LAZZER, F. 2010. A Metamodel for Software Requirement Patterns. *REFSQ 2010, LNCS 6182* Berlin Heidelberg: Springer-Verlag.

FROLICK, M. & ARIYACHANDRA, T. 2006. Business performance management: One truth. *Information Systems Management,* 23**,** 41–48.

GACITUA, R., MA, L., NUSEIBEH, B., PIWEK, P., DE ROECK, A., ROUNCEFIELD, M., SAWYER, P., WILLIS, A. & YANG, H. Making Tacit Requirements Explicit. Second International Workshop on Managing Requirements Knowledge (MARK), 2009 Atlanta, Georgia, USA. IEEE, 40-44.

GALLIERS, R. D. & LAND, F. F. 1987. Choosing Appropriate Information Systems Research Methodologies. *Communications of the ACM,* 30**,** 900-902.

GAMMA, E., JOHNSON, R., HELM, R. & VLISSIDES, J. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson Education.

GARLAN, D., ALLEN, R. & OCKERBLOOM, J. 1995. Architectural mismatch: Why reuse is so hard. *IEEE software,* 12**,** 17-26.

GARVER, M. S. & MENTZER, J. T. 1999. Logistics research methods: Employing structural equation modeling to test for construct validity. *Journal of Business Logistics,* 20**,** 33-57.

GAŠEVIĆ, D., KAVIANI, N. & MILANOVIĆ, M. 2009. Ontologies and software engineering. *Handbook on Ontologies***,** 593-615.

GAVRILOVA, T. 2010. Knowledge Engineering for Non-Engineers. *Intelligent Information Processing V - IFIP Advances in Information Technology and Communication Technology.* Springer.

GENNARI, J. H., MUSEN, M. A., FERGERSON, R. W., GROSSO, W. E., CRUBÉZY, M., ERIKSSON, H., NOY, N. & TU, S. W. 2003. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies,* 58**,** 89-123.

GIBSON, M. & CONHEENEY, K. 1995. Domain knowledge reuse during requirements engineering. *Advanced Information Systems Engineering, Lecture Notes in Computer Science.* Springer Berlin Heidelberg.

GIBSON, M. & SAEEDI, M. A Knowledge Based Approach to Requirements Engineering. 3rd European Conference on Information Systems, 1995 Athens, Greece. 277-291.

GMATI, I., RYCHKOVA, I., BARRIOS, J. & NURCAN, S. Return on Experience of the Implementation of a Business-IT alignment Approach: Theory and Practice. 5th International Workshop on Business/IT Alignment and Interoperability, 2010 Hammamet, Tunisia. CEUR Workshop Proceedings, 76-90.

GOEDVOLK, J. G. 1999. Integrated Architecture Framework. Amsterdam.

GOLDIN, L. & BERRY, D. 2013. Reuse of requirements reduced time to market at one industrial shop: a case study. *Requirements Engineering. Requirements Engineering,* 20**,** 23-44.

GOMAA, H., KERSCHBERG, L., SUGUMARAN, V., BOSCH, C., TAVAKOLI, I. & O'HARA, L. 1996. A Knowledge-Based Software Engineering Environment for Reusable Software Requirements and Architectures. *Automated Software Engineering 3***,** 285-307.

GOMAA, H. & OLIMPIEW, E. 2008. Managing Variability in Reusable Requirement Models for Software Product Lines. *High Confidence Software Reuse in Large Systems, Lecture Notes in Computer Science,* 5030**,** 182-185.

GOMES, P. & BENTO, C. Learning User Preferences in Case-Based Software Reuse. 5th European Workshop on Advances in Case-Based Reasoning, 2000 Trento, Italy. 112-123.

GOMEZ-PEREZ, A., FERNÁNDEZ-LOPEZ, M. & CORCHO, O. 2004. *Ontological Engineering,* Springer.

GONZALEZ, A. & BARR, V. 2000. Validation and verification of intelligent systems—what are they and how are they different? *Journal of Experimental & Theoretical Artificial Intelligence***,** 407–420.

GRANT, R. 1996. Toward A Knowledge-Based Theory of the Firm. *Strategic Management Journal***,** 109-122.
GREGOR, S., HART, D. & MARTIN, N. 2007. Enterprise architectures: enablers of business strategy and IS/IT alignment in government. *Information Technology & People,* 20**,** 96–120.
GRONAU, N. & ROHLOFF, M. Managing change: business/IT alignment and adaptability of information systems. ECIS 07, 2007 Geneva, Switzerland. AIS, 1741-1753.
GRUBER, T. R. 1992. Ontolingua: A mechanism to support portable ontologies. *Knowledge Systems Laboratory*.
GRUBER, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge acquisition,* 5**,** 199-220.
GUARINO, N. Formal ontology in information systems. Formal Ontology in Information Systems, 1998. IOS Press, 3–15.
GUARINO, N. & WELTY, C. 2000. A formal ontology of properties. *Knowledge Engineering and Knowledge Management Methods, Models, and Tools***,** 97-112.
GUPTA, U. 1993. Validation and Verification of Knowledge-Based Systems: A Survey. *Journal of Applied Intelligence,* 3**,** 343-363.
GUTIERREZ, A. & SERRANO, A. 2008. Assessing strategic, tactical and operational alignment factors for SMEs: alignment across the organisation's value chain. *International Journal of Value Chain Management,* 2**,** 33-56.
GUTIERREZ, A., OROZCO, J., SERRANO, A. & SERRANO, A. Using tactical and operational factors to assess strategic alignment: An SME study. European and Mediterranean Conference on Information Systems, 2006 Costa Blanca, Alicante, Spain. EMCIS, 1-10.
HAES, S., HAEST, R. & VAN GREMBERGEN, W. 2010. IT Governance and Business-IT Alignment in SMEs. *Journal of Information Systems Control,* 6**,** 1-7.
HAES, S. & VAN GREMBERGEN, W. 2004. IT Governance and Its Mechanisms. *Journal of Information Systems Control,* 1**,** 27-33.
HAES, S. & VAN GREMBERGEN, W. 2005. Measuring and Improving IT Governance. *Journal of Information Systems Control,* 2**,** 35-42.
HAES, S. & VAN GREMBERGEN, W. 2009a. An Exploratory Study into IT Governance Implementations and its Impact on Business/IT Alignment. *Information Systems Management,* 26**,** 123–137.
HAES, S. & VAN GREMBERGEN, W. 2009b. Exploring the relationship between IT governance practices and business/IT alignment through extreme case analysis in Belgian mid-to-large size financial enterprises. *Journal of Enterprise Information Management,* 22**,** 615-637.
HAIGH, M. 2010. Software quality, non-functional software requirements and IT-business alignment. *Software Quality Journal,* 18**,** 361-385.
HAPPEL, H. & SEEDORF, S. Applications of Ontologies in Software Engineering. International Workshop on Semantic Web Enabled Software Engineering, 2006 Athens, GA, U.S.A.: Springer, 1-14.
HARDY, G. 2006. Using IT governance and COBIT to deliver value with IT and respond to legal, regulatory and compliance challenges. *Information Security Technical Report II***,** 55-61.

HARSU, M. 2002. A Survey of Domain Engineering. *Tampere University of Technology* [Online]. Available: http://practise2.cs.tut.fi/pub/papers/domeng.pdf.

HEIDARI, F., LOUCOPOULOS, P. & BRAZIER, F. Ontology for quality specification in requirements engineering. 4th International Conference on Models and Ontology-based Design of Protocols, Architectures and Services, 2013. 7-12.

HENDERSON, J. & VENKATRAMAN, N. 1993. Strategic alignment: leveraging information technology for transforming organization. *IBM Systems Journal***,** 472-484.

HEVNER, A. & CHATTERJEE, S. 2010. Design Research in Information Systems. *Integrated Series in Information Systems,* 2**,** 9-22.

HEVNER, A. R., MARCH, T. S., PARK, J. & SUDHA, R. 2004. Design science in information systems research. *MIS Quarterly,* 28**,** 75–105.

HOELTER, D. R. 1993. The analysis of covariance structures: Goodness-of-fit indices. *Sociological Methods and Research,* 11**,** 325–344.

HOFFMANN, M., KUHN, N., WEBER, M. & BITTNER, M. Requirements for requirements management tools. 12th IEEE International Conference on Requirements Engineering, 2004. IEEE, 301-308.

HOMOD, S. & RINE, D. 1999. Building requirements Repository Using Requirements Transformation Techniques to Support Requirements Reuse. *Cybernetics and Informatics,* 2.

HOOD, C., WIEDEMANN, S., FICHTINGER, S. & PAUTZ, U. 2007. *Requirements Management: the interface between requirements development and all other systems engineering processes,* Springer Science & Business Media.

HORRIDGE, M., DRUMMOND, N., GOODWIN, J., RECTOR, A., STEVENS, R. & WANG, H. 2006. The Manchester OWL Syntax. OWL: Experiences and directions. University of Manchester.

HOSSEINBEIG, S., KARIMZADGAN-MOGHADAM, D., VAHDAT, D. & MOGHADAM, R. A. IT strategic alignment maturity and IT governance. 4th International Conference on Interaction Sciences, 2011. 67-72.

HU, Q. & HUANG, C.D. 2006. Using The Balanced Scorecard To Achieve Sustained IT-Business Alignment: A Case Study. *Communications of the Association for Information Systems,* 17**,** 181-204.

HUANG, C. D. & HU, Q. 2007. Achieving IT-business strategic alignment via enterprise-wide implementation of balanced scorecards. *Information Systems Management,* 24**,** 173-184.

HUGOSON, M. & PESSI, K. 2011. Operational and Structural Business IT Alignment. *BIS 2011 Workshops, LNBIP 97.* Berlin Heidelberg: Springer-Verlag.

JACKSON, D. L., GILLASPY JR, J. A., & PURC-STEPHENSON, R. 2009. Reporting practices in confirmatory factor analysis: an overview and some recommendations. *Psychological methods,* 14**,** 6-23.

JAHNKE, A. 2004. Why Is Business-IT Alignment So Difficult? *CIO-FRAMINGHAM MA-,* 17**,** 78-80.

JARVENPAA, S. L. & IVES, B. 1994. Organizational Fit and Flexibility: IT Design Principles for a Globally Competing Firm. *Strategy, Organization Design and Human Resource Management,* 3**,** 1-39.

JIANG, L., EBERLEIN, A. & FAR, B. 2008. A case study validation of a knowledge-based approach for the selection of requirements engineering techniques. *Requirements Engineering,* 13**,** 117–146.

JOHNSON, A. & LEDERER, A. 2010. CEO/CIO mutual understanding, strategic alignment, and the contribution of IS to the organization. *Information and Management,* 41**,** 138-149.

JONES, C. 1997. *Software quality analysis and guidelines for success,* New York, International Thomson Computer Press.

JRAD, R., AHMED, M. & SUNDARAM, D. Insider Action Design Research a multi-methodological Information Systems research approach. IEEE Eighth International Conference on Research Challenges in Information Science, 2014. IEEE Press, 1-12.

KAIYA, H. & SAEKI, M. Using domain ontology as domain knowledge for requirements elicitation. 14th IEEE International Conference on Requirements Engineering, 2006. IEEE Press, 189-198.

KAPLAN, R. 2008. Conceptual foundations of the balanced scorecard. *Handbooks of Management Accounting Research,* 3**,** 1253-1269.

KAPLAN, R. & NORTON, D. 1992. The balanced scorecard as a strategic management system. *Harvard Business Review***,** 61–66.

KAPLAN, R. & NORTON, D. 1996. *The Balanced Scorecard-Translating Strategy into Action,* Boston, Harvard Business School Press.

KARPOVSKY, A. & GALLIERS, R. 2015. Aligning in practice: from current cases to a new agenda. *Journal of Information Technology,* 30**,** 136-160.

KASHANCHI, R. & TOLAND, J. 2006. Can ITIL contribute to IT/business alignment? An initial investigation. *Wirtschaftsinformatik,* 48**,** 340-348.

KEARNS, G. & LEDERER, A. 2003. A Resource-Based View of Strategic IT Alignment: How Knowledge Sharing Creates Competitive Advantage. *Decision Sciences,* 34**,** 1-29.

KENDAL, S. & CREEN, M. 2007. *An Introduction to Knowledge Engineering,* London, Springer-Verlag.

KHAIATA, M. & ZUALKERNAN, I. A. 2009. A simple instrument to measure IT-business alignment maturity. *Information Systems Management,* 26**,** 138-152.

KITAMURA, M., HASEGAWA, R., KAIYA, H. & SAEKI, M. A supporting tool for requirements elicitation using a domain ontology. Software and Data Technologies, 2009. Springer Berlin Heidelberg, 128-140.

KNAUSS, E. & EL BOUSTANI, C. Assessing the Quality of Software Requirements Specifications. Conference on International Requirements Engineering, 2008 Barcelona, Spain. IEEE Press, 341-342.

KOF, L., GACITUA, R., ROUNCEFIELD, M. & SAWYER, P. Ontology and Model Alignment as a Means for Requirements Validation. 4th International Conference on Semantic Computing, 2010 Pittsburgh, USA. IEEE, 46-51.

KOSSMANN, M. & ODEH, M. Ontology-driven requirements engineering–A case study of OntoREM in the aerospace context. 20th Annual International

Symposium of the International Council on Systems Engineering, 2010 Chicago, IL, USA. INCOSE, 1000-1013.

KOSSMANN, M., WONG, R., ODEH, M. & GILLIES, A. Ontology-driven Requirements Engineering: Building the OntoREM Meta Model. International Conference on Information and Communication Technologies: From Theory to Applications, 2008. IEEE Press, 1-6.

KOZAKI, K., SUNAGAWA, E., KITAMURA, Y. & MIZOGUCHI, R. Distributed Construction of Ontologies Using Hozo. Workshop on Social and Collaborative Construction of Structured Knowledge colocated with WWW2007, 2007 Banff.

KUJALA, S., KAUPPINEN, M., LEHTOLA, L. & KOJO, T. The role of user involvement in requirements quality and project success. Proceedings of 13th IEEE International Conference on Requirements Engineering, 2005. 75-84.

LAFFEY, T., COX, P., SCHMIDT, J., KAO, S. & READ, J. 1988. Real-Time Knowledge-Based Systems. *AI Magazine,* 9**,** 27-45.

LAM, W., MCDERMID, J. A. & VICKERS, A. J. 1997. Ten Steps Towards Systematic Requirements Reuse. *Requirements Engineering,* 2**,** 102-113.

LAMSWEERDE, A. & LETIER, E. 2000. Handling Obstacles in Goal-Oriented Requirements Engineering. *IEEE Transactions in Software Engineering,* 26**,** 978-1005.

LAMSWEERDE, A., LETIER, E. & DARIMONT, R. 1998. Managing Conflicts in Goal-Driven Requirements Engineering. *IEEE Transactions on Software Engineering,* 24**,** 908-926.

LAPOUCHNIAN, A. 2005. Goal-oriented requirements engineering: An overview of the current research. Toronto: University of Toronto.

LEDERER, A. & MENDELOW, A. 1986. Issues in Information Systems Planning. *Information and Management,* 10**,** 245-254.

LEE, R. 2013. *Software Engineering: A Hands-On Approach* Atlantis Press.

LEE, Y. & ZHAO, W. Domain requirements elicitation and analysis-An ontology-based approach. 2006. Springer Berlin Heidelberg, 805-813.

LÉGER, A., HEINECKE, J., NIXON, L. J., SHVAIKO, P., CHARLET, J., HOBSON, P. & GOASDOUÉ, F. 2009. Semantic web take-off in a european industry perspective. *Semantic Web for Business: Cases and Applications.* Hershey, PA: IGI Global.

LEI, P. & WU, Q. 2007. Introduction to Structural Equation Modelling: Issues and Practical Considerations, An NCME Instructional Module. *Instructional Topics in Educational Management***,** 33-43.

LEINWEBER, D. 1988. Knowledge-Based Systems for Financial Applications. *IEEE Expert***,** 18-31.

LEWIS, G., MORRIS, E., O'BRIEN, L., SMITH, D. & WRAGE, L. 2005. SMART: The service-oriented migration and reuse technique. Carnegie-Mellon University, Pittsburgh, Software Engineering Institute.

LI, C., POOLEY, R. & LIU, X. 2010. Ontology-Based Quality Attributes Prediction. *International Journal of Computer Science & Information Technology,* 2**,** 12-29.

LIBERATORE, M. & STYLIANOU, A. 1994. Using knowledge-based systems for strategic market assessment. *Information & Management,* 27**,** 221–232.

LIN, J., FOX, M. S. & BILGIC, T. 1996. A requirement ontology for engineering design. *Concurrent Engineering,* 4**,** 279-291.

LIU, C. L. Ontology-based conflict analysis method in non-functional requirements. 9th International Conference on computer and Information Science, 2010. IEEE Press, 491-496.

LOPEZ DE MANTARAS, R., MCSHERRY, D., BRIDGE, D., LEAKE, D., SMYTH, B., CRAW, S., FALTINGS, B., MAHER, M., COX, M., FORBUS, K., KEANE, M., AAMODT, A. & WATSON, I. 2005. Retrieval, Reuse, Revision and Retention in Case-based Reasoning. *Journal of Knowledge Engineering,* 20**,** 215-240.

LOUCOPOULOS, P. & CHAMPION, R. 1988. Knowledge-based approach to requirements engineering using method and domain knowledge. *Knowledge-Based Systems,* 1**,** 179–187.

LOUCOPOULOS, P. & CHAMPION, R. 1989. Knowledge-Based Support for Requirements Engineering. *Journal of Information Software Technology,* 31**,** 123-135.

LUFTMAN, J. 2000. Assessing Business-IT alignment maturity. *Communications of Association of Information Systems,* 4**,** 99-111.

LUFTMAN, J. 2003. Assessing IT/Business Alignment. *Information Systems Management,* 20**,** 9-15.

LUFTMAN, J., BEN-ZVI, T., DWIVEDI, R. & RIGONI, E. H. 2010. IT Governance: An Alignment Maturity Perspective. *International Journal of IT/Business Alignment and Governance,* 1**,** 13-25.

LUFTMAN, J. & BRIER, T. 1999. Achieving and Sustaining Business-IT Alignment. *California Management Review,* 42**,** 109-122.

LUFTMAN, J., DOROCIAK, J., KEMPAIAH, R. & RIGONI, E. H. Strategic alignment maturity: a structural equation model validation. AMCIS 2008 Proceedings, Paper 53, 2008 Toronto, Canada. AIS at http://aisel.aisnet.org/amcis2008/53/, 1-16.

LUFTMAN, J., PAPP, R. & BRIER, T. 1999. Enablers and Inhibitors of Business–IT Alignment. *Communications of the Association for Information Systems***,** 1–33.

LUFTMAN, J. & RAJKUMARK, K. 2007. An update on business/alignment: a line has been drawn. *MIS Quarterly Executive,* 6**,** 165-177.

MAALEJ, W. & THURIMELLA, A. K. 2013. An Introduction to Requirements Knowledge. *Managing Requirements Knowledge.* Springer-Verlag Berlin Heidelberg.

MACCALLUM, R. C., BROWNE, M. W. & SUGAWARA, H. M. 1996. Power analysis and determination of sample size for covariance structure modeling. *Psychology Methods,* 1**,** 130–149.

MACEHITER, N. & WARD-DUTTON, N. 2005. On IT-business alignment. *MWD Advisors* [Online]. Available: www.mwdadvisors.com [Accessed February 19].

MAES, R. 1999. Reconsidering information management through a generic framework. Amsterdam: Universiteit of Amsterdam.

MAES, R., RIJSENBRIJ, D., TRUIJENS, O. & GOEDVOLK, H. 2000. Redefining Business – IT alignment Through a Unified Framework. Universiteit Van Amsterdam/CapGemini.

MAIDEN, N. & SUTCLIFFE, A. Reuse of analogous specifications during requirements analysis. 6th International Workshop on Software Specification and Design, 1991. IEEE Press, 220-223.
MAIDEN, N. & SUTCLIFFE, A. 1993a. Case-Based Reasoning in Software Engineering, IEE Colloquium on Case-Based Reasoning. *IEE Colloquium on Case-Based Reasoning***,** 2/1-2/3
MAIDEN, N. & SUTCLIFFE, A. 1993b. Exploiting reusable specification through analogy. *Communications of the ACM,* 35**,** 55-64.
MALLICK, S. & KRISHNA, S. Requirements engineering: problem domain knowledge capture and deliberation process support. International Workshop on Database and Expert Systems Applications, 1999. 392-397.
MARRONE, M. & KOLBE, L. M. 2011. Uncovering ITIL claims: IT executives' perception on benefits and Business-IT alignment. *Information Systems and E-Business Management,* 9**,** 363-380.
MASSONET, P. & VAN LAMSWEERDE, A. 1997. Analogical reuse of requirements frameworks. *Third IEEE International Symposium on Requirements Engineering***,** 26-37.
MATSATSINIS, N., DOUMPIS, M. & ZOPOUNIDIS, C. 1997. Knowledge Acquisition and Representation for Expert Systems in the Field of Financial Analysis. *Expert Systems with Applications,* 12**,** 247-262.
MAYANK, V., KOSITSYNA, N. & AUSTIN, M. 2004. Requirements Engineering and the Semantic Web - Part II, Representation, Management, and Validation of Requirements and System-Level Architectures. University of Maryland.
MCCLURE, C. 1997. *Software Reuse Techniques*, Prentice-Hall.
MCDONALD, R. P. & HO, M. H. R. 2002. Principles and practice in reporting structural equation analyses. *Psychological methods,* 7**,** 64-82.
MCGOWAN, J. 2010. The Rise of Computerized High Frequency Trading: Use and Controversy. *Duke Law & Technology Review,* 16**,** 16-19.
MCKEEN, J. & SMITH, H. 2003. *Making IT Happen: Critical Issues in IT Management,* Kingston, Wiley.
MCLEOD, L. & MACDONELL, S. G. 2011. Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR),* 43**,** 24-56.
MENG, Y., YOUNG, A., WANG, X. & SHAO, K. Domain-Analysis in Software Reuse - Application in Warehouse Management. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. 505-510.
MILI, A., MILI, R. & MITTERMEIR, R. T. 1998. A survey of software reuse libraries. *Annals of Software Engineering,* 5**,** 349-414.
MIRIYALA, K. & HARANDI, M. T. 1991. Automatic derivation of formal software specifications from informal descriptions. *IEEE Transactions on Software Engineering,* 17**,** 1126-1142.
MONTEALEGRE, R. 2002. A Process Model of Capability Development: Lessons from the Electronic Commerce Strategy at Bolsa de Valores de Guayaquil. *Organization Science,* 13**,** 514-531.
MORISIO, M., EZRAN, M. & TULLY, C. 2002. Success and failure factors in software reuse. *IEEE Transactions on Software Engineering,* 28**,** 340–357.

MOTJOLOPANE, I. & BROWN, I. Strategic business-IT alignment, and factors of influence: a case study in a public tertiary education institution. 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries, 2004. South African Institute for Computer Scientists and Information Technologists, 147-156.

MUELLER, R. O. & HANCOCK, G. R. 2008. Best Practices in Structural Equation Modeling. *Best Practices in Quantitative Methods.* Thousand Oaks, California: Sage.

NAKASHIMA, T., ARIYAMA, T., KITANO, H. & ISHIBUCHI, H. 2005. A Fuzzy Rule-Based Trading Agent: Analysis and Knowledge Extraction. *Studies in Computational Intelligence,* 2**,** 265-277.

NGUYEN, T., VO, B., LUMPE, M. & AND GRUNDY, J. REInDetector: A Framework for Knowledge-based Requirements Engineering. 27th IEEE/ACM International Conference on Automated Software Engineering, 2012 Essen, Germany. 386-389.

NGUYEN, T., VO, B., LUMPE, M. & GRUNDY, J. 2014. KBRE: a framework for knowledge-based requirements engineering. *Software Quality Journal***,** 87–119.

NIU, N., SAVOLAINEN, J., NIU, Z., JIN, M. & CHENG, J., (2014) 2014. A Systems Approach to Product Line Requirements Reuse. *IEEE Systems Journal,* 8**,** 827-836.

NORTHROP, L. & CLEMENTS, C. 2012. A Framework for Software Product Line Practice, Version 5.0. *Software Engineering Institute (SEI)* [Online]. Available: http://www.sei.cmu.edu/productlines/frame_report.

NOY, N. & MCGUINNESS, D. 2005. Ontology Development 101: A Guide to Creating Your First Ontology. *protegewiki* [Online]. Available: http://protegewiki.stanford.edu/wiki/Ontology101.

NUNAMAKER, J., CHEN, M. & PURDIN, T. 1991. Systems Development in Information Systems Research. *Journal of Management Information Systems,* 7**,** 89-106.

NUSEIBEH, B. & EASTERBROOK, S. Requirements engineering: a roadmap. Conference on the Future of Software Engineering, 2000 New York, USA. ACM, 35-46.

OMG 2014. Financial Industry Business Ontology (FIBO)

OMORONYIA, I., SINDRE, G., STÅLHANE, T., BIFFL, S., MOSER, T. & SUNINDYO, W. 2010. A domain ontology building process for guiding requirements elicitation. *Requirements Engineering: Foundation for Software Quality***,** 188-202.

PALMER, J. & MARKUS, M. 2000. The performance impacts of quick response and strategic alignment in specialty retailing. *Information Systems Research,* 11**,** 241-259.

PALMER, J. & MYERS, M. 1988. Knowledge-based Systems Application to Reduce Risks in Software Engineering. *Proceedings of 2nd International Conference on Uncertainty and Intelligent Systems.* Urbino, Italy: Springer.

PAPP, R. 1999. Business-IT Alignment: Productivity Paradox Payoff? *Industrial Management and Data Systems,* 99**,** 367-373.

PAUL, D., YEATES, D. & CADLE, J. 2010. *Business Analysis (2ed),* Swindon, UK, British Informatics Society Limited.

PEPPARD, J. & BRUE, K. Beyond Alignment: A Coevolutionary View of the Information Systems Strategy Process. International Conference on Information Systems, 2003 Seattle, USA. 743-750.

PEPPARD, J., GALLIERS, R. D. & THOROGOOD, A. 2014. Information Systems as Practice: Micro strategy and strategizing for IS. *The Journal of Strategic Information Systems,* 23**,** 1-10.

PETERAF, M. A. 1993. The cornerstones of competitive advantage: a resource-based view. *Strategic management journal,* 14**,** 179-191.

PFAHL, D. & LEBSANFT, K. 2000. Using simulation to analyse the impact of software requirement volatility on project performance. *Information and Software Technology,* 42**,** 1001-1008.

PRESSMAN, R. S. 2005. *Software engineering: a practitioner's approach,* New York, Palgrave Macmillan.

PRESTON, D. & KARACHANNA, E. 2006. Antecedents of IS Strategic Alignment: A Nomological Network. *Information Systems Research,* 20**,** 159-179.

QRUNFLEH, S. & TARAFDAR, M. Tactical IT-business alignment: antecedents and outcomes. 39th Annual Meeting of the Decision Sciences Institute, 2008 Baltimore, USA. Decision Sciences Institute, Baltimore, 3331-3336.

RADLINSKI, L. 2012. Empirical Analysis of the Impact of Requirements Engineering on Software Quality. *Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science Volume 7195.* Springer.

RALHA, C. & GOSTINSKI, R. 2008. A methodological framework for business-IT alignment. *Business-IT alignment strategies: a conceptual modelling.* Salvador IEEE.

REGEV, G., FAVRE, J., HAYEK, E., WILSON, P. & WEGMANN, A. Business/IT Alignment in Practice: Lessons Learned from a Requirements Project at P&G. Advanced Information Systems Engineering Workshops, 2011. Springer Berlin Heidelberg, 93-101.

REICH, H. & BENBASSAT, I. 1996. Measuring the linkage between business and information technology objective. *MIS Quarterly,* 20**,** 55-81.

REICH, H. & BENBASSAT, I. 2000. Factors That Influence the Social Dimension of Alignment between Business and information Technology Objective. *MIS Quarterly,* 24**,** 81-114.

REINHARTZ-BERGER, I., SOFFER, P. & STURM, A. 2005. A Domain Engineering Approach to Specifying and Applying Reference Models. *EMISA,* 75**,** 50-63.

ROBERTSON, S. & ROBERTSON, J. 2012. *Mastering the Requirements Process: Getting Requirements Right 3e,* Boston, MA, Addison-wesley.

RUNESON, P. & HÖST, M. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering,* 14**,** 131-164.

RYAN, K. & MATHEWS, B. Matching conceptual graphs as an aid to requirements re-use. IEEE International Symposium on Requirements Engineering, 1993. IEEE, 112-120.

SABHERWAL, R. & CHAN, Y. E. 2001. Alignment between Business and IS Strategies: A study of prospectors, analysers, and defenders. *Information Systems Research,* 12**,** 11-33.
SABHERWAL, R., HIRSCHHEIM, R. & GOLES, T. 2001. The Dynamics of Alignment: Insignts from a Punctuated Equilibrium Model. *Organization Science,* 12**,** 179-197.
SALGADO, C., MACHADO, R. J. & SUZANA, R. Modeling the alignment between business and IS/IT: a requirements engineering perspective. 28th Annual ACM Symposium on Applied Computing, 2013. ACM, 1309-1311.
SAMANTA, S. Business-IT alignment strategies: a conceptual modelling. Proceeding of PICMET, 2007. 1-5.
SCHLOSSER, F. Mastering the Social IT/Business Alignment Challenge. AMCIS 2012 Proceedings, 2012 Seattle, USA. AIS, 1843-1849.
SCHLOSSER, F., BEIMBORN, D., WEITZEL, T. & WAGNER, H. 2015. Achieving social alignment between business and IT- an empirical evaluation of the efficacy of IT governance mechanisms. *Journal of Information Technology,* 30**,** 119-135.
SCHREIBER, J. B., NORA, A., STAGE, F. K., BARLOW, E. A. & KING, J. 2006. Reporting structural equation modeling and confirmatory factor analysis results: A review. *Journal of educational research,* 99**,** 323-338.
SCHWABER, C. & STERPE, P. 2007. *Selecting The Right Requirements Management Tool—Or Maybe None Whatsoever.* Cambridge, USA: Forrester Research, Inc.
SCHWALBE, K. 2014. *Information Technology Project Management, Revised,* Boston, MA, CENGAGE Learning.
SCOTT, W. & COOK, S. C. An architecture for an intelligent requirements elicitation and assessment assistant. 13th Annual International Symposium Proceedings of INCOSE, 2003. 470-479.
SEAMAN, C. B. 1999. Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering,* 25**,** 557-173.
SELBY, R. 2005. Enabling reuse-based software development of large-scale systems. *IEEE Transactions on Software Engineering,* 31**,** 495-514.
SEPEHRI, M., ABDOLLAHZADEH, A., SEPEHRI, M. & GOODARZI, M. Impact of Quality Factors on the Success of Software Development. European Computing Conference, 2009. 523-532.
SHARMA, A., KUMAR, R. & GROVER, P. S. 2007. A Critic Survey of Reusability Aspects for Component-Based Systems. *Proceedings of World Academy of Science, Engineering and Technology,* 2i.
SHAW, M. & GAINES, R. 1996. Requirements Acquisition. *IEE Software Engineering Journal,* 11**,** 149-165.
SHERIF, K. & VINZE, A. 2003. Barriers to adoption of software reuse: a qualitative study. *Information & Management,* 41**,** 159-175.
SHIBAOKA, M., KAIYA, H. & SAEKI, M. GOORE: Goal-oriented and ontology driven requirements elicitation method. Advances in Conceptual Modeling–Foundations and Applications, 2007. Springer Berlin Heidelberg, 225-234.

SHIUE, W., LI, S. & CHEN, K. 2008. A frame knowledge system for managing financial decision knowledge. *Expert Systems with Applications,* 35**,** 1068–1079.

SIEGEMUND, K., THOMAS, E. J., ZHAO, Y., PAN, J. & ASSMANN, U. Towards ontology-driven requirements engineering. 10th International Semantic Web Conference Workshop on Semantic Web Enabled Software Engineering, 2011 Bonn, Germany. Springer.

SILVIUS, A. Business & IT alignment in theory and practice. 40th Annual International Conference on System Sciences, 2007 Hawaii. IEEE Press, 211-220.

SILVIUS, A. J., DE WAAL, B. & & SMIT, J. 2009. Business and IT alignment; answers and remaining questions. Pacific Asia Conference on Information Systems (PACIS 2009), 2009 Hyderabad, India. AIS, 44.

SIMONIN, J., LE TRAON, Y. & JÉZÉQUEL, J. M. An Enterprise Architecture Alignment Measure for Telecom Service Development. 11th IEEE International Enterprise Distributed Object Computing Conference, 2007 Shanghai, China IEEE, 476-483.

SIMONSEN, J. 1999. How do we take care of strategic alignment? *Scandinavian Journal of Information Systems***,** 51-72.

SJØBERG, D. I., DYBÅ, T., ANDA, B. C. & HANNAY, J. E. 2008. Building theories in software engineering. *Guide to advanced empirical software engineering.* London: Springer.

SLEDGIANOWSKI, D., LUFTMAN, J., REILLY & R. 2006. Development and Validation of an Instrument to Measure Maturity of IT Business Strategic Alignment Mechanisms. *Information Resources Management Journal,* 19**,** 18-33.

SMACZNY, T. 2001. Is an alignment between business and information technology the appropriate paradigm to manage IT in today's organisations? *Management Decision,* 39**,** 797-802.

SODHI, J. & SODHI, P. 1999. *Software Reuse: Domain Analysis and Design Process Computing,* McGraw-Hill.

SOLAIMANI, S. & DE VRIES, E. Towards a Human Processual Approach of Business-IT Alignment. 16th American Conference on Information Systems, 2010 Lima, Peru. AIS, 50.

SOMMERVILLE, I. 2011. *Software Engineering 9e,* Boston, Addison-Wesley.

SOMMERVILLE, I., RODDEN, T., SAWYER, P., BENTLEY, R. & TWIDALE, M. Integrating ethnography into the requirements engineering process. IEEE International Symposium on Requirements Engineering, 1993. IEEE, 165-173.

SOMMERVILLE, I. & SAWYER, P. 1997a. *Requirements Engineering- A Good Practice Guide,* Chichester, UK, John Wiley.

SOMMERVILLE, I. & SAWYER, P. 1997b. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering 3***,** 101-130.

STRNADL, C. Aligning Business and IT: The Process-Driven Architecture Model. International Conference on Computer as a Tool, EUROCON 2005, 2005 Belgrade, Serbia & Montenegro. IEEE Press, 1048 – 1051.

SUHR, D. D. Exploratory or confirmatory factor analysis? SAS Users Group International Conference, 2006 Cary, NC. SAS Institute, 1-17.
SUNAGAWA, E., KOZAKI, K., KITAMURA, Y. & MIZOGUCHI, R. 2004. Organizing role-concepts in ontology development environment: Hozo. The Institute of Scientific and Industrial Research, Osaka University.
SURE, Y., STAAB, S. & STUDER, R. 2004. On-to-knowledge methodology (OTKM). *Handbook on ontologies*, 117-132.
SURE, Y., STAAB, S. & STUDER, R. 2009. Ontology engineering methodology. *Handbook on ontologies*, 135-152.
SUTCLIFFE, A. Scenario-based Requirements Engineering. 11th IEEE International Conference on Requirements Engineering, 2003. IEEE, 320-329.
SUTCLIFFE, A. & MAIDEN, N. 1998. The domain theory for requirements engineering. *IEEE Transactions on Software Engineering,* 24**,** 174-196.
SUTCLIFFE, A., MAIDEN, N., MINOCHA, S. & MANUEL, D. 1998. Supporting Scenario-Based Requirements Engineering. *IEEE Transactions on Software Engineering,* 24**,** 1072-1088.
SUTCLIFFE, A. & SAWYER, P. Requirements Elicitation: Towards the Unknown Unknowns. 21st IEEE International Requirements Engineering Conference (RE), 2013 IEEE. Rio de Janeiro, Brazil, 92-104.
SYMONS, C. 2005. IT and business alignment: Are we there yet. Forrester Research, Inc.
TALLON, P. & KRAEMER, K. 2003. Investigating the Relationship between Strategic Alignment and IT Business Value: The discovery of a paradox. *Creating Business Value with Information Technology: Challenges and Solutions.* Hershey, PA: Idea Group Publishing.
TALLON, P. & PINOSONNEAULT, A. 2011. Competing Perspectives on the Link between Strategic Alignment and Organizational Agility: Insights from a mediation model. *MIS Quarterly,* 35**,** 463-486.
TAMAI, T. & KAMATA, M. 2009. Impact of Requirements Quality on Project Success or Failure, Design Requirements Engineering: A Ten-Year Perspective. *Lecture Notes in Business Information Processing,* 14**,** 258-275.
TARAFDAR, M. & QRUNFLEH, S. 2009. IT-business alignment: a two-level analysis. *Information Systems Management,* 26**,** 338-349.
TARTIR, S., ARPINAR, I. B., MOORE, M., SHETH, A. P. & ALEMAN-MEZA, B. OntoQA: Metric-based ontology quality analysis. IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, 2005.
TEECE, D. J. 2007. Explicating Dynamic Capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal,* 28**,** 1319-1350.
TEECE, D. J., PISANO, P. & SHUEN, A. 1997. Dynamic Capabilities and Strategic Management. *Strategic Management Journal,* 18**,** 509-533.
TEO, T. & ANG, J. 1999. Critical Success Factors in the Alignment of IS Plans with Business plans. *International Journal of Information Management,* 19**,** 173-185.

TIAN, J. 2005. *Software Quality Engineering - Testing, Quality Assurance, and Quantifiable Improvement,* Hoboken, New Jersey, John Wiley & Sons, Inc.
TRACZ, W. 1988. *Software reuse: emerging technology,* IEEE Computer Society Press.
TRIPATHI, K. A Review on Knowledge-based Expert System: Concept and Architecture. IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches & Practical Applications, 2011. 19-23.
TSAI, W., JIN, Z., WANG, P. & WU, B. Requirement Engineering in Service-Oriented System Engineering. IEEE International Conference on e-Business Engineering, 2007 IEEE Press. Hong Kong, 661-668.
TSOUMAS, B., DRITSAS, S. & GRITZALIS, D. 2005. An ontology-based approach to information systems security management. *Computer Network Security***,** 151-164.
TSUI, F., KARAM, O. & BERNAL, B. 2014. *Essentials of Software Engineering 3e,* Burlington, USA, Jones & Bartlett Learning.
TURBAN, E. & ARONSON, J. 2001. *Decision Support Systems and Intelligent Systems 6th ed,* Upper Saddle River, NJ, Prentice Hall.
ULLAH, A. & LAI, R. 2011. Modeling business goal for business/it alignment using requirements engineering. *Journal of Computer Information Systems,* 51**,** 21-28.
ULLAH, A. & LAI, R. 2013. Requirements Engineering and Business/IT Alignment: Lessons Learned. *Journal of Software,* 8**,** 1-10.
ULLMAN, J. B. 2006. Structural equation modeling: Reviewing the basics and moving forward. *Journal of Personality Assessment,* 87**,** 35-50.
UMOH, E. & SAMPAIO, P. The 'REFINTO' Framework and Tool: Supporting Business-IT Alignment in Enterprise Financial Application Development. 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 Ulm, Germany. IEEE Press, 406-409.
UMOH, E., SAMPAIO, P. & THEODOULIDIS, B. REFINTO: An ontology-based requirements engineering framework for business-IT alignment in financial services organizations. 8th IEEE Conference on Service Computing, 2011 Washington DC. IEEE Press, 600-607.
UMOH, E., SAMPAIO, P. & THEODOULIDIS, B. 2012. Measuring and Evaluating Business-IT Alignment for RAD Projects Using the REFINTO Framework and Tool. *Practice-Driven Research on Enterprise Transformation: 4th Working Conference, PRET 2012, Gdańsk, Poland, June 27, 2012, Proceedings.* Gdańsk, Poland: Springer-Verlag Berlin Heidelberg.
VAISHNAVI, V. & KUECHLER, W. 2007. *Design Science Research Methods and Patterns: Innovating Information and Communication Technology,* New York, Auerbach Publications, Taylor & Francis Group.
VAN DER ZEE, J. & DE JONG, B. 1999. Alignment Is Not Enough: Integrating Business and Information Technology Management with the Balanced Business Scorecard. *Journal of Management Information Systems,* 16**,** 137-156.
VAN GREMBERGEN, W., HAES, S. & GULDENTOPS, E. 2003. Structures, Processes and Relational Mechanisms for IT Governance. *Strategies for*

Information Technology Governance. Hersey, USA: PA Ideal Group Publishing.
VAN GREMBERGEN, W. & SAULL, R. Aligning business and information technology through the balanced scorecard at a major Canadian financial group: its status measured with an IT BSC maturity model. 34th Annual International Conference on System Sciences, 2001 Hawaii, USA. IEEE Press, 10-19.
VENKATRAMAN, N. 1989. The concept of fit in strategy research: toward verbal and statistical correspondence. *The Academy of Management Review,* 14**,** 423-444.
VERES, C., SAMPSON, J., COX, K., BLEISTEIN, S. & VERNER, J. 2010. An Ontology-Based Approach for Supporting Business-IT Alignment. *Complex Intelligent Systems and Their Applications***,** 21-42.
VESSEY, I. & WARD, K. 2013. The Dynamics of Sustainable IS Alignment: The Case for IS Adaptivity. *Journal of the Association of Information Systems,* 14**,** 283-311.
VICAT, C., BREZILLON, P. & NOTTOLA, C. 1995. Knowledge validation in the building of a knowledge-based system, Expert Systems with Applications. *European Verification and Validation of Knowledge-Based Systems,* 8**,** 391–397.
VIEIRA, A. 2011. *Interactive LISREL in Practice: Getting Started with a SIMPLIS Approach,* London, Springer Heidelberg.
VLIET, H. 2007. *Software Engineering: Principles and Practice*, Wiley.
WANG, X., ZHOU, X. & JIANG, L. A Method of Business and IT Alignment Based on Enterprise Architecture. 2008. 740-745.
WEILL, P. & BROADBENT, M. 1998. Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology. Boston: Harvard Business School Press.
WEISS, D. & LAI, C. 1999. *Software Product-Line Engineering: A Family-Based Software Development Process*, Addison-Wesley.
WERNERFELT, B. 1984. A resource-based view of the firm. *Strategic management journal,* 5**,** 171-180.
WHITTINGTON, R. 2014. Information Systems Strategy and Strategy-as-Practice: A joint agenda. *The Journal of Strategic Information Systems,* 23**,** 87-91.
WIEGERS, K. & BEATTY, J. 2013. *Software Requirements - Best Practices, 3e,* Redmond, Washington, Microsoft Press.
WILSON, B. & WELSH, J. 1986. Small knowledge-based systems in education and training: Something new under the sun. *Educational Technology,* 26**,** 7-13.
WINTER, S. G. 2003. Understanding dynamic capabilities. *Strategic Management Journal,* 24**,** 991-995.
WITHALL, S. 2007. *Software Requirement Patterns (Developer Best Practices),* Washington, Microsoft Corporation.
WOHLIN, C. & AURUM, A. 2014. Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering Journal,* 1**,** 1-29.
WOHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M. C. & REGNELL, B. W. A. 2012. *Experimentation in Software Engineering,* Springer.

WOOLFE, R. 1993. The Path to Strategic Alignment. *Information Strategy: The Executive's Journal,* 9**,** 13-23.

WOUTERS, B., DERIDDER, D. & VAN PAESSCHEN, E. The Use of Ontologies as a Backbone for Use Case Management. European Conference on Object-Oriented Programming (ECOOP 2000), Workshop : Objects and Classifications, a natural convergence, 2000 Cannes, France Springer, 1-6.

YANG, Y., XIA, F., ZHANG, W., XIAO, X., LI, Y. & LI, X. Towards semantic requirement engineering. IEEE International Workshop on Semantic Computing and Systems, 2008. IEEE Press, 67-71.

YAYLA, A. & HU, Q. Antecedents and Drivers of IT-Business Strategic Alignment: Empirical Validation of a Theoretical Model. European Conference on Information Systems, 2009 Verona, Italy. 158-169.

YAYLA, A. A. & HU, Q. 2012. The Impact of IT-Business Strategic Alignment on Firm Performance in a Developing Country Setting: Exploring moderating roles of environmental uncertainty and strategic orientation. *European Journal of Information Systems,* 21**,** 373-387.

YU, E. S. Towards modelling and reasoning support for early-phase requirements engineering. Third IEEE International Symposium on Requirements Engineering, 1997 Annapolis, MD, USA. IEEE, 226-235.

ZAVE, P. & JACKSON, M. 1997. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering Methodology,* 6**,** 1-30.

ZOLLO, M. & WINTER, S. G. 2002. Deliberate learning and the evolution of dynamic capabilities. *Organization Science,* 13**,** 339-353.

ZOPOUNIDIS, C., DOUMPIS, M. & MATSATSINIS, N. 1997. On the use of knowledge-based decision support systems in financial management: A survey. *Decision Support Systems,* 20**,** 259-277.

ZOWGHI, D. & COULIN, C. 2005. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. *Engineering and Managing Software Requirements.*

ZOWGHI, D. & NURMULIANI, N. A Study of the Impact of Requirements Volatility on Software Project Performance. Asia-Pacific Software Engineering Conference, 2002 Queensland, Australia. IEEE Press, 3-11.

List of Associated Publications

1. UMOH, E. & SAMPAIO, P. The 'REFINTO' Framework and Tool: Supporting Business-IT Alignment in Enterprise Financial Application Development. 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 Ulm, Germany. IEEE Press, 406-409.

2. UMOH, E., SAMPAIO, P. & THEODOULIDIS, B. 2012. Measuring and Evaluating Business-IT Alignment for RAD Projects Using the REFINTO Framework and Tool. Practice-Driven Research on Enterprise Transformation: 4th Working Conference, PRET 2012, Gdańsk, Poland, June 27, 2012, Proceedings. Gdańsk, Poland: Springer-Verlag Berlin.

3. UMOH, E., SAMPAIO, P. & THEODOULIDIS, B. REFINTO: An ontology-based requirements engineering framework for business-IT alignment in financial services organizations. 8th IEEE Conference on Service Computing, 2011 Washington DC. IEEE Press, 600-607.

4. UMOH, E., SAMPAIO, P. & THEODOULIDIS, B. Forces Influencing Business-IT Alignment: A practical approach and empirical study (Journal Paper - pending review)

Appendix A: Implementation of REFINTO Framework Support Tool

A1: RAMM Presentation Layer Implementation

The presentation layer of the RAMM module is developed in FXML[12] using JavaFX[13] and Scene Builder[14]. The FXML developed in Scene Builder was edited in IntelliJ IDE[15] was used for building the control program and linking the FXML (view) to the controller and model. The IntelliJ IDE interface is shown in Figure A1.

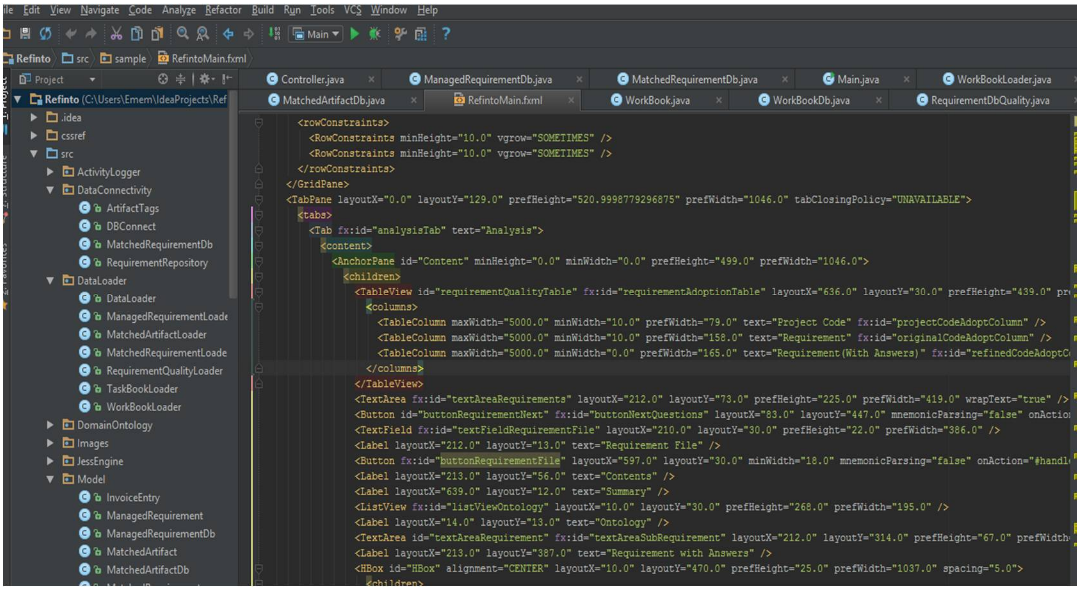

Figure A1: RAMM Presentation Layer development in IntelliJ IDE

The main components of the presentation layer are GUI controls for making selections such of project type, category, stage, and ontology files to load. There are toolbars that aid navigation through the elicitation, refinement, and persistence workflow. There are tabs and data grids that show statuses and facilitate responding to questions generated from the service layer (knowledge engine).

---

[12] A tutorial on FXML is available at: http://docs.oracle.com/javafx/2/api/javafx/fxml/doc-files/introduction_to_fxml.html
[13] JavaFX is an open source tool from Oracle shipped with Java SDK
[14] Scene Builder is the visual layout tool for JavaFX
[15] IntelliJ Community Edition is available at: https://www.jetbrains.com/idea/download/

A2: RAMM Service Layer Implementation

The service or business logic layer serves a number of functions which can be grouped into loading functions, display functions, and logical and reasoning functions. The software codes that implements these functions are highlighted in this appendix. The loading function involves reading requirement documents and the domain ontology files into RAMM, parsing the content into file stream readers and traversing the trees and nodes of the domain ontologies. An extract of the Java code for the requirements loading function in IntelliJ IDE is shown in Figure A2.

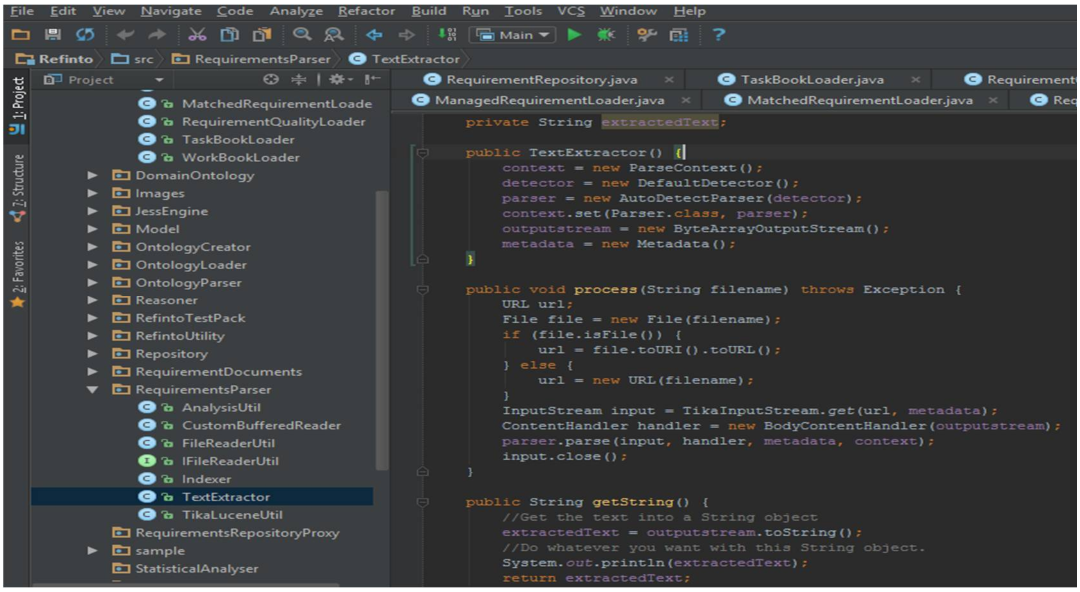

Figure A2: Extract of RAMM Requirements Parser function

The requirement documents can be in file formats like Microsoft Word (DOC) or text (TXT). The requirements loading classes use Apache Tika API for detection and extraction of metadata and text from various file types. Tika offers other advanced functions such as search engine indexing, content analysis and translation. Other classes within the RAMM service layers like workbook loader and task loaders function by calling into the data layer components to retrieve, process, and serve data to the presentation layer.

The REFINTO framework support tool workflow is driven by the categorization of the project being analysed and the corresponding domain ontologies. The ontology loading function is therefore critical. The ontology loading classes in RAMM utilizes functions in OWLAPI. The API provides a number of classes and interfaces like OWLDataFactory, OWLOntology, OWLOntologyManager, and OWLManager to facilitate creation, loading and saving of ontologies.

The snippet of code that facilitates the loading of an already developed ontology such as reconciliation ontology is shown in Listing A1:

Listing A1: Ontology Loading

```
OWLOntologyManager oMgr = create();
OWLOntology o = oMgr.loadOntologyFromOntologyDocument(reconciliation_iri);
assertNotNull(o);
```

An extract of the ontology loading class in RAMM is show in Figure A3.

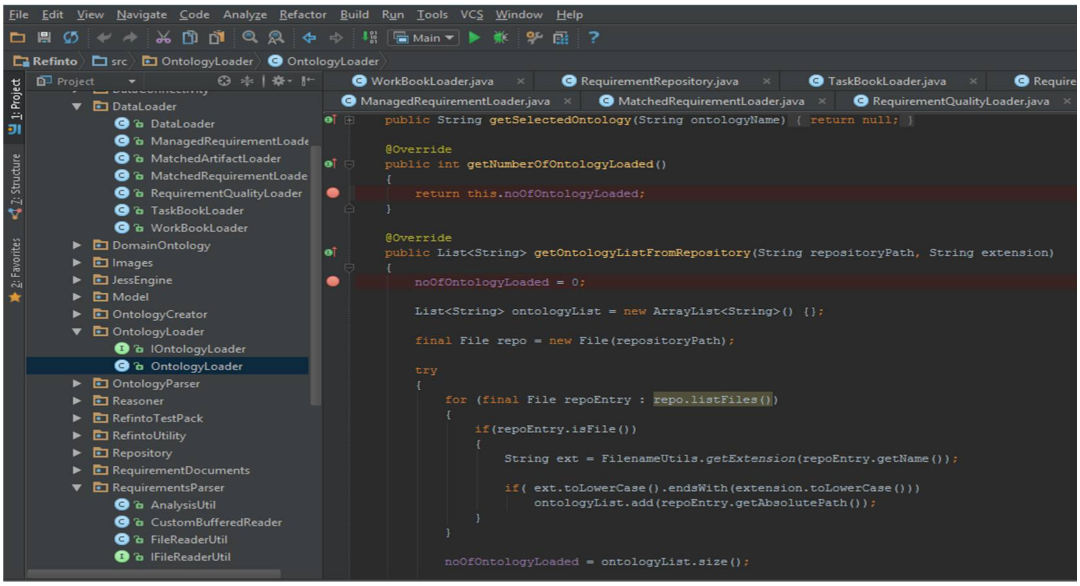

Figure A3: Extract of RAMM Requirements Parser function

The RAMM service layer also provides logical and reasoning functions using OWLAPI, OWLOntology, and OWLOntologyWalker classes and interfaces. In the context of requirement elicitation, ontology tree walking is used to raise questions that are related to the business function domain ontology. OWLAPI also provides classes and interfaces that facilitate searching for restrictions and annotations which were useful for this purpose. Using the reconciliation domain ontology for example, the ontology is traversed and based on this questions are presented on the user interface. This process is used to elicit and refine requirements. The code snippet for tree walking is shown in Listing A2.

Listing A2: Ontology Tree Walking:

```
OWLOntologyManager oMgr = create();
OWLOntology o = oMgr.loadOntologyFromOntologyDocument(reconciliation_iri);
OWLOntologyWalker walker = new OWLOntologyWalker(Collections.singleton(o));
OWLOntologyWalkerVisitor<Object> visitor =
new OWLOntologyWalkerVisitor<Object>(walker) {
@Override
public Object visit(OWLObjectSomeValuesFrom desc) {
        System.out.println(desc);
        System.out.println(" " + getCurrentAxiom());
        return null;
        }
};
walker.walkStructure(visitor);
```

For the reasoning functions, the OWLAPI classes and interfaces used were OWLReasoner and OWLReasonerFactory as shown in code snippet in Listing A3.

Listing A3: Ontology Reasoning:

```
OWLOntologyManager m = create();
OWLOntology o = oMgr.loadOntologyFromOntologyDocument(reconciliation_iri);
ReasonerProgressMonitor progressMonitor = new LoggingReasonerProgressMonitor(
LOG, "testDescendants");
OWLReasonerConfiguration config = new SimpleConfiguration(progressMonitor);
OWLReasoner reasoner = reasonerFactory.createReasoner(o, config);
reasoner.precomputeInferences(InferenceType.CLASS_HIERARCHY);
for (OWLClass c : o.getClassesInSignature()) {
assert c != null;
NodeSet<OWLClass> subClasses = reasoner.getSubClasses(c, true);
        for (OWLClass subClass : subClasses.getFlattened()) {
        assertNotNull(subClass);
                        subClassProcessing(subClass);
        }
}
```

An extract from the service layer model code in IntelliJ is shown in Figure A4.

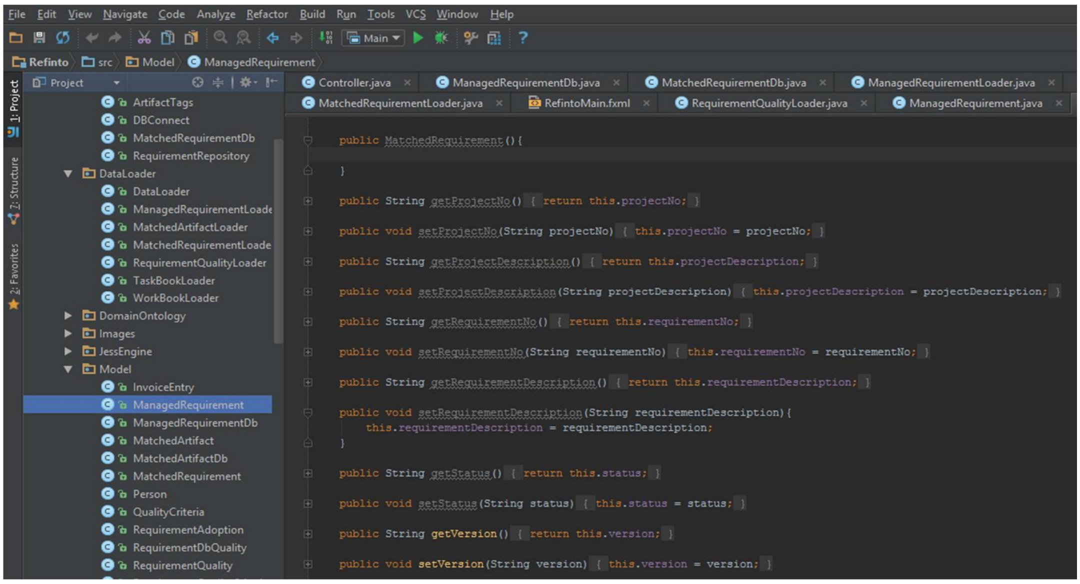

Figure A4: RAMM Service Layer Model

A3: RAMM Data Layer Implementation

RAMM data layer is made up of the knowledge base and shared data components. The knowledge base is made up of the REFINTO framework domain ontologies, rules, and cased based components. The domain ontologies are held offline in a file directory and loaded in on demand. The RAMM knowledge base is made up of rules and cases based components as depicted in Figure A5.

The rule-based aspect of the knowledge base is developed in JESS. The case-based aspects of the knowledgebase rely on the shared repository for persistence and retrieval. The cases can be serialized into XML, Protobuf[16], or JSON[17]. There is interaction between the case and rule sets.

The functions of the rules component include rule definition, representation of the cases in a format that can be consumed in JESS, and applying rules to identify hidden features. It also responsible for representing the serialized case retrieved from the repository in any of the three formats (XML, Protobuf, or JSON) identified that can be consumed by the case component.

[16] Google's Interchange Format available at: https://github.com/google/protobuf
[17] JavaScript Object Notation is a lightweight data-interchange format: http://json.org/

To build the rules based reasoning templates, rules and facts have to be defined. Templates define concepts or objects in the domain of interest through a set of properties. Facts refer to instantiation of objects or concepts defined in the templates. JESS templates are analogous to classes in Java classes, each having a name like Java classes. Errors are thrown if an attempt is made to access a rule or fact that has not been defined in a template. A sample template defined for the reconciliation ontology is provided in Listing A4.

Listing A4: Rule Template:

```
 (deftemplate Reconciliation_Requirement
   Reconciliation Requirement Sec
        (slot Reconciliation_Data)
        (slot Reconciliation_System)
        (slot Reconciliation_ProcessOutcome)
        (slot Reconciliation_Recipient)
        (slot Reconciliation_StorageMedia)
        (slot Reconciliation_DeliveryMedia)
        (slot Reconciliation_RunMode)
)
```

Rules were defined for the templates following an if-then pattern. For the reconciliation template in Listing A4, a snippet of the rule defined in JESS format is provided in Listing A.5.

Listing A5: Rule Template:

```
(defrule reconciliation_rule_example
        (Reconciliation Requirement (Run Mode "Automated") =>
        (System.out.println("The Reconciliation Requirement Run Mode is Automated")
)
```

Facts can then be added into the working memory using assert as shown in Listing A6.

Listing A6: Using of Assert to Add a Fact into Working Memory

```
(assert (Reconciliation Requirement
(Data "Transactional Data")
(System "Equities Trading System")
(Process Outcome "Success")
(Recipient "Senior Traders")
(Storage Media "Shared Directory")
(Delivery Media "Email")
(Run Mode "Automated")))
```

With the facts added to using asset becomes available in the working memory for the reasoning process and can be executes using the JESS keyword run. Following the representation of knowledge in the form of rules, the inference engine determines the rule which is executed for a given fact. The JESS inference engine is uses Rete 1 algorithm which is an efficient pattern matching algorithm for implementation of production rule systems. Code snippet illustrating the use of the JESS rule engine is provided in Listing A7.

Listing A7: Using JESS Rule Engine

```
Rete engine = new Rete();
try {
	engine.executeCommand("(batch \"Template.clp\")");
	engine.executeCommand("(batch \"newCase.clp\")");
}
catch(Exception e){
	e.printStackTrace();
}
```

A4: DEMM Presentation Layer Implementation

The implementation of the presentation layer of the DEMM module is based on XAML. Visual Studio[18], an IDE from Microsoft for .NET development, was used to develop the XAMLs. The IDE has a built-in support for interactive design interface feature similar to JavaFx's Scene Builder. The IDE seamlessly integrates the XAML and code flies. Various XAMLs were created for the various user controls and views as shown in Figure A5.

Similar to the RAMM module, the main components of the DEMM module presentation layer are controls for making selections. There are tool and menu bars that also support easy navigation through the workflow of the DEMM module which includes viewing reports, performing evaluations and monitoring the status of projects both on the grids and pictorially in charts. The statistical analysis and aggregated alignment scores for the selected projects can be viewed. This was done through using

[18] Visual Studio is an IDE for .NET Development

appropriate third party open source libraries and built-in controls within Visual Studio.

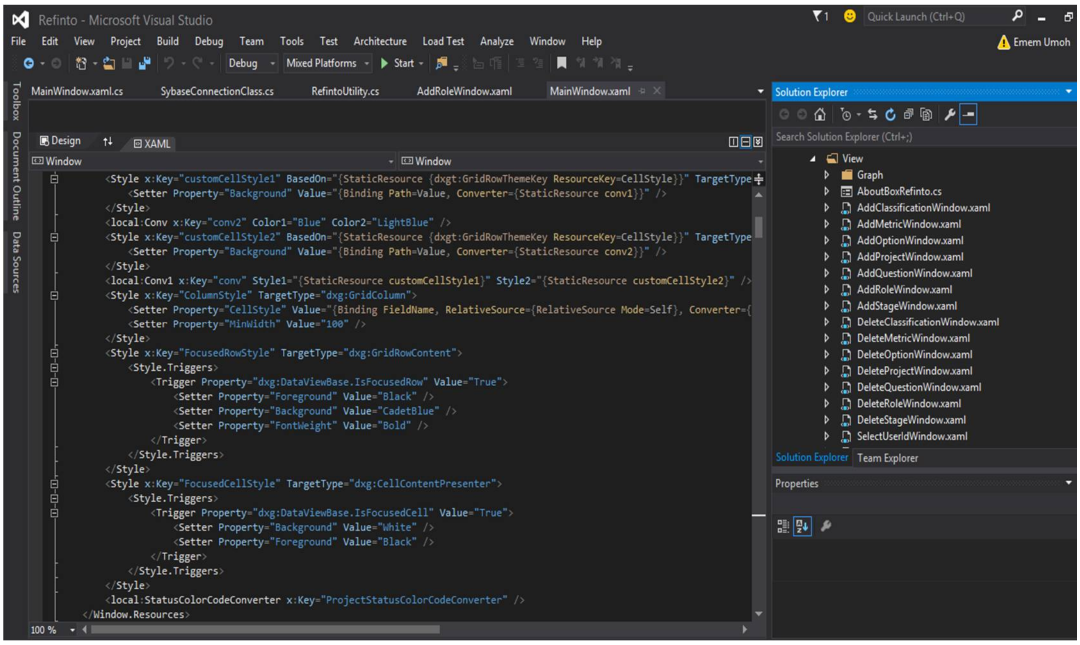

Figure A5: DEMM Presentation Layer Extract

A5: DEMM Service Layer Implementation

The display function in DEMM is built in WPF and uses MVVM architectural constructs to drive interaction between the view and view model through notifications, data binding and commands. The Controls in the view are bound to the properties and commands exposed by the view model through the *DataContext* property. Value converters were created and used on the view to format data for display on the user interface. Validation rules were used in some cases to provide user input validation.

An extract of code from the DEMM service layer is shown in Figure A6.

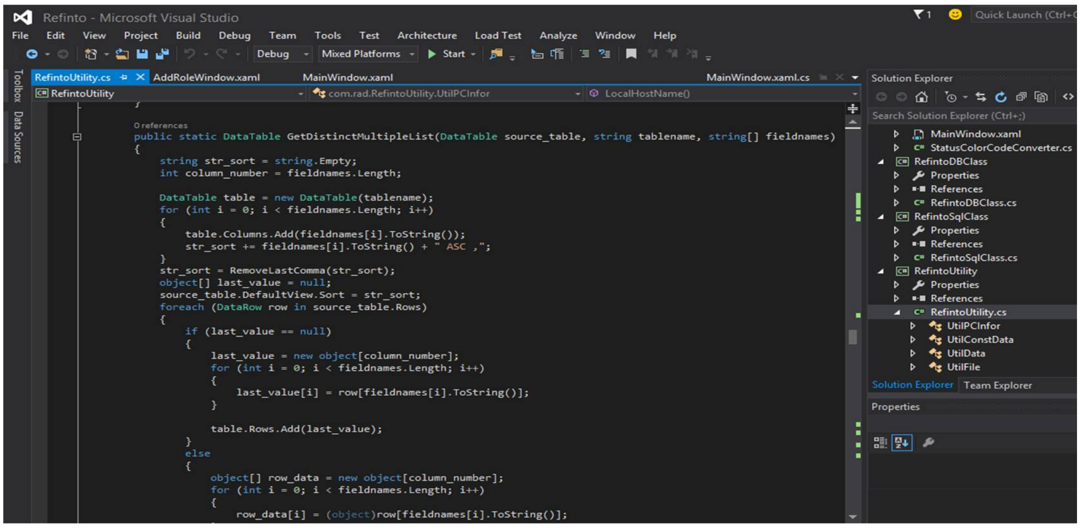

Figure A6: DEMM Service Layer Code Extract

The view model classes are non-visual class and do not derive from WPF base classes. In line with the MVVM pattern, view models were written to encapsulate the presentation logic for user interaction. The view models were independently testable from the view. Properties and commands data bound to the view were defined in the from the models. The notification of changes in the view model to reflect in the view is achieved by implementing the *INotifyPropertyChanged* interfaces. A code snippet that performs notification is provided in Listing A8

Listing A8: Property Implementation with INotifyPropertyChanged

```
private string _ ProjectClassification;
public string ProjectClassification
{
  get
  {
          return _ ProjectClassification;
  }
  set
  {
          _ ProjectClassification = value;
          OnPropertyChanged(new PropertyChangedEventArgs("ProjectClassification"));
  }}
```

Implementation of a command in the service layer is shown in the code snippet in Listing A9. This is achieved by implementing the *ICommand* interface.

Listing A9: Command Implementation

```
private ICommand runAnalyisisCommand (Umoh et al., Umoh and Sampaio)
public ICommand RunAnalyisisCommand
{
        get
        {
          return runAnalyisisCommand;
        }
        set
        {
          runAnalyisisCommand = value;
        }
}
```

The statistical analysis functions developed makes use of freely available statistical library for Microssoft.NET from FoundaSoft called *FoundaStat*. The REFINTO statistical analysis classes are wrappers for calls to computation capabilities of the FoundaStat library such as descriptive statistics, correlation analysis, statistical inferences, ANOVA, regression analysis, and six sigma analysis. An extract of the statistical analysis classes using FoundaStat is depicted in Figure A7.

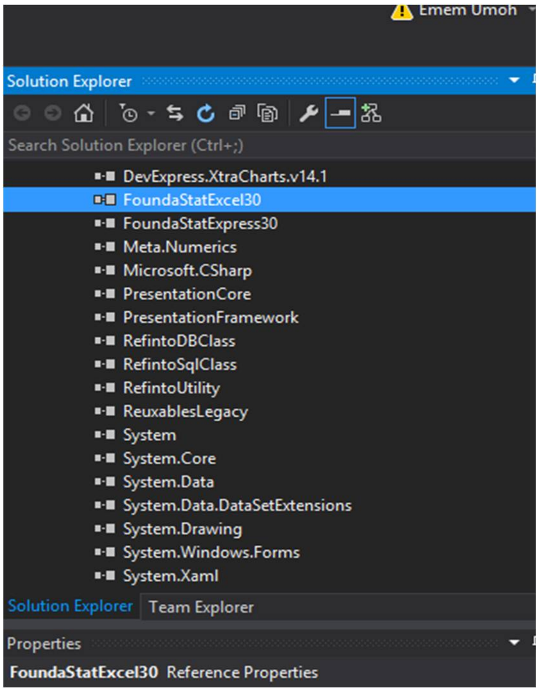

Figure A7: DEMM FoundaStat Reference

Code extract showing the wrapping of the FoundaStat API is shown in Figure A8.

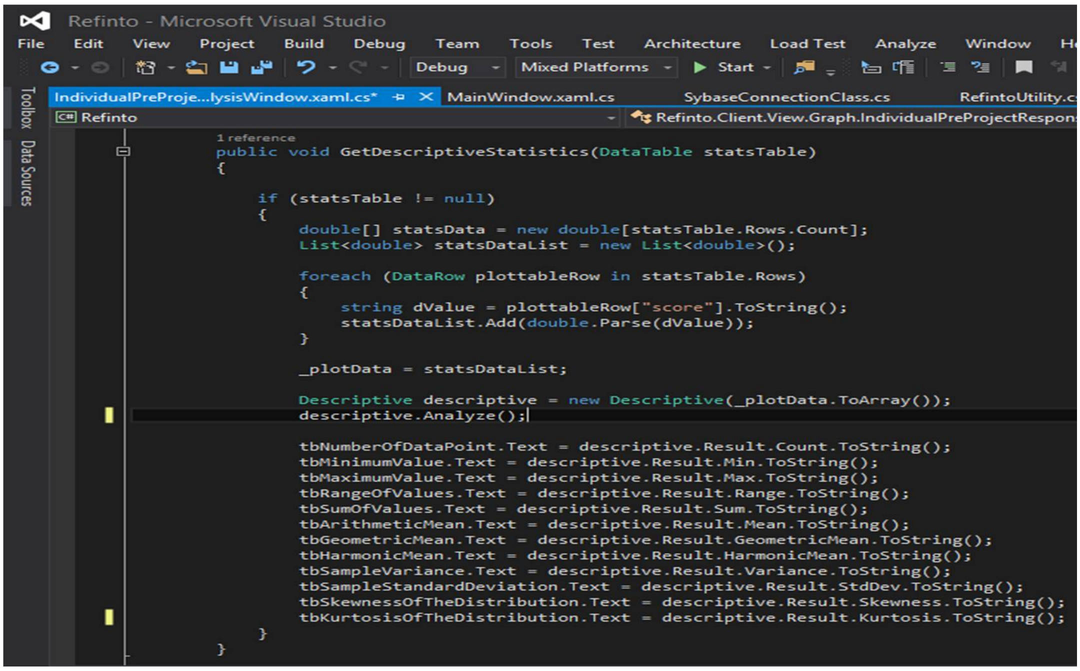

Figure A8: DEMM Statistical Analysis Code Extract

A6: DEMM Service Layer Implementation

The data layer of DEMM calls into stored procedures that are created for the REFINTO framework. This layer also provided classes that perform the key functions of data retrieval and persistence. The classes written for the data layer is based on the ADO.NET[19] and Sybase APIs.

The main classes in the ADO.NET API are *DataSet*, and *DataProvider*. DataSet objects represent a disconnected cache of data. This is made up of *DataTables* and *DataRelations* which represent the result of commands. The commands in the case of the DEMM data layer classes contain the stored procedures.

[19] ADO.NET is a set of APIs for accessing data and data services based on disconnected DataSets and XML

In Figure A9, code snippet of the DEMM data layer classes showing is shown. The results of the calls are processed in the service layer and the updates reflected on the presentation layer through the notification constructs already discussed.

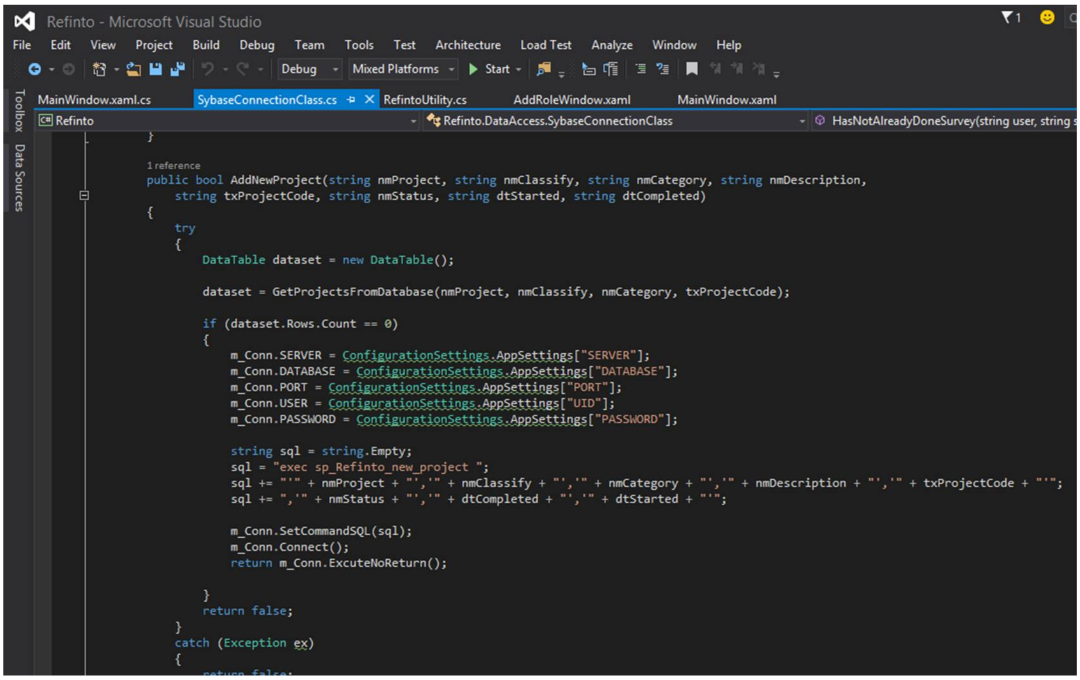

Figure A9: DEMM Data Layer Code Extract

A7: Implementation of Shared Components

The database management system (DBMS) used for REFINTO framework support tool is ASE Sybase 15 Developer Edition[20]. To implement this layer, tables had to be created in using a DBMS. Stored procedures which wrap the queries that perform the CRUD (Create, Read, Update and Delete) functions in the DEMM module were written.

Stored procedures offer an advantage over inline SQL in code. These include improved maintainability, easier testing, isolation of business rules, improved execution speed, and optimization, and security.

[20]ASE Sybase Developer Edition from SAP is available at: http://scn.sap.com/docs/DOC-46595

The shared infrastructure is made up of 23 tables and 31 stored procedures as shown in Figure A10.

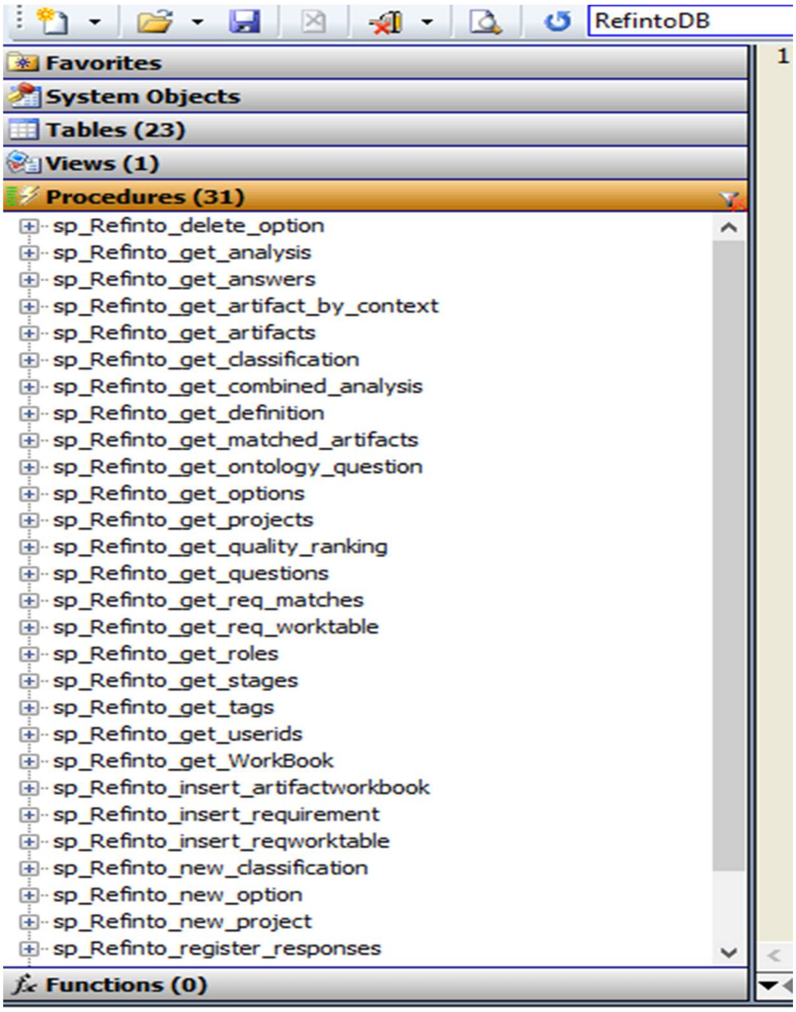

Figure A10: REFINTO Shared Component

Appendix B: REFINTO Domain Ontology Representation in Protégé and UML

Common Ontologies:

I. Delivery Media Ontology

The delivery media ontology represents the medium by which the output of a process such as reporting and reconciliation can be conveyed to interested parties, represented in another common ontology (Recipient ontology).

The output being conveyed has a format, represented in a common ontology (Data Format ontology). The delivery media ontology can be useful in the requirements elicitation process by making knowledge of available delivery media options explicit. The delivery media ontology is depicted in Figure B1. The classes and individuals are defined in Table B1.

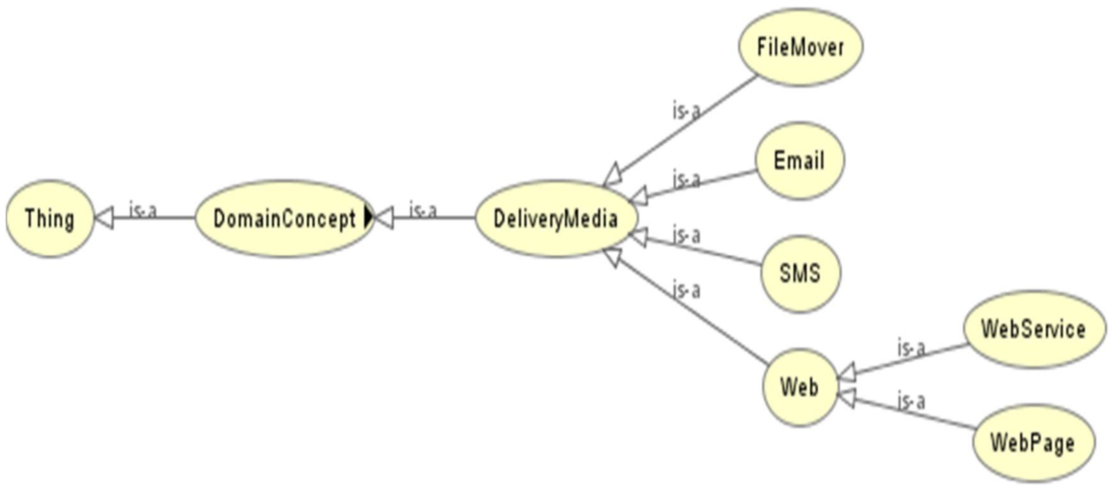

Figure B1: Extract of Delivery Media Ontology

Table B1: Delivery Media Ontology Definitions

Term	Definition	Type	Synonym
Delivery Media	Medium through which the output of process such as data or Report is conveyed to interested parties	Class	
Web	Web media to convey process output to interested parties accessed through a Web browser	Class	
Web Service	Service via transport protocol such as SOAP and HTTP	Individual	
Web Page	Page available on Intranet or Internet	Individual	
SMS	Mobile text message for conveying process output	Individual	Text
Email	Electronic mail as a medium for conveying process output	Individual	
File Mover	Utility for moving files into Storage Media	Individual	

II. Storage Media Ontology

There is a need to have output of processes such as reconciliation and reports persisted in some form for various reasons such as future litigations investigations. This can be stipulated by regulatory requirement or organizational policy. Storage media serves this function.

The storage media ontology is shown in Figure B2. During requirements elicitation the storage media ontology is useful for making knowledge of available storage media options explicit. Restrictions (constraints) on the use of a storage media can also be captured and made explicit through the ontology.

The storage media ontology classes and individuals are defined in Table B2.

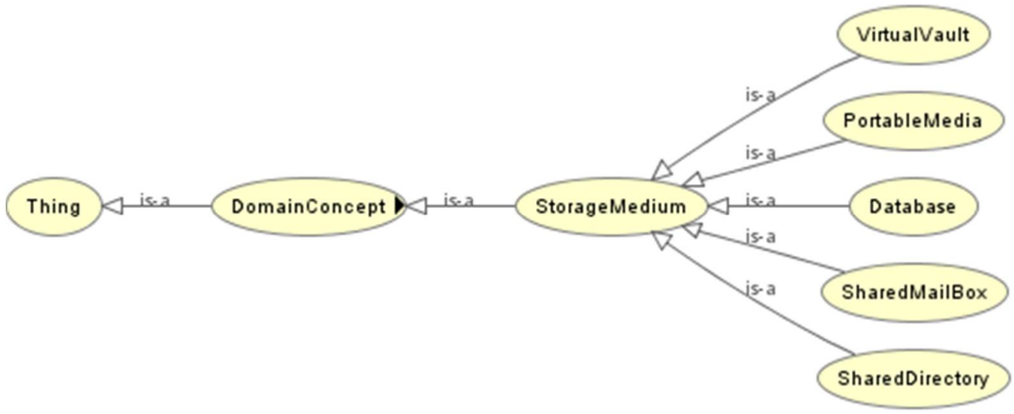

Figure B2: Extract of Storage Media Ontology

Table B2: Storage Media Ontology Definitions

Term	Definition	Type	Synonym
Storage Media	Media for Persisting output of process	Class	
Portable Media	Storage media designed for mobility	Class	
Virtual Vault	Digital data storage with advanced security features	Individual	
Database	Database management system with tables for storing data	Class	
Shared Mailbox	Group mailbox typically shared by a business team	Individual	
Shared Directory	File Share Directory on a network drive	Individual	

III. Retention Ontology

The output of a process and associated data (reference or transactional) has to be retained for a period of time. The period of time can be hours, days, weeks, months or years as captured in the RetentionPeriod class. The reason for retention, captured in the RetentionReason class could be business need, regulatory stipulation or organizational policy. The Retention ontology is depicted in Figure B3. The Retention ontology classes and individuals are defined in Table B3.

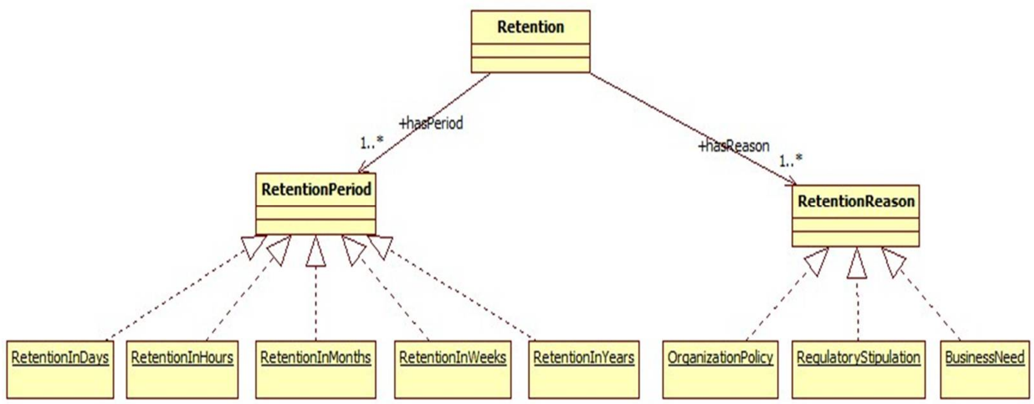

Figure B3: Extract of Retention Ontology in UML

Table B3: Retention Ontology Definitions

Term	Definition	Type	Synonym
Retention	Persisting output of process such as data or Report	Class	
Retention Period	Period for which output of process such as data is kept	Class	
has Period	Retention is for a period of time	Property	
Retention In Days	Retention period specified in days	Individual	
Retention In Hours	Retention period specified in hours	Individual	
Retention In Weeks	Retention period specified in weeks	Individual	
Retention In Months	Retention period specified in months	Individual	
Retention In Years	Retention period specified in years	Individual	
Retention Reason	Reason for which Retention is required	Class	
has Reason	Retention has a reason	Property	
Business Need	Business requires Retention based on the procedures	Individual	
Regulatory Stipulation	Retention is required by Regulator for specified period	Individual	
Organization Policy	Retention is stipulated in organization's policy	Individual	

IV. Data Format Ontology

Data format (file format or data type) format refers to the encoding of data for storage in a computer file system. The traditionally used file formats include XLS (MS Excel file), DOC (MS Word file). It can also refer to serializable formats such as Protobuf, XML and JSON, consumed mostly by machine agents which deserialize and present it in other formats. Extract of the data format ontology is depicted in Figure B4.

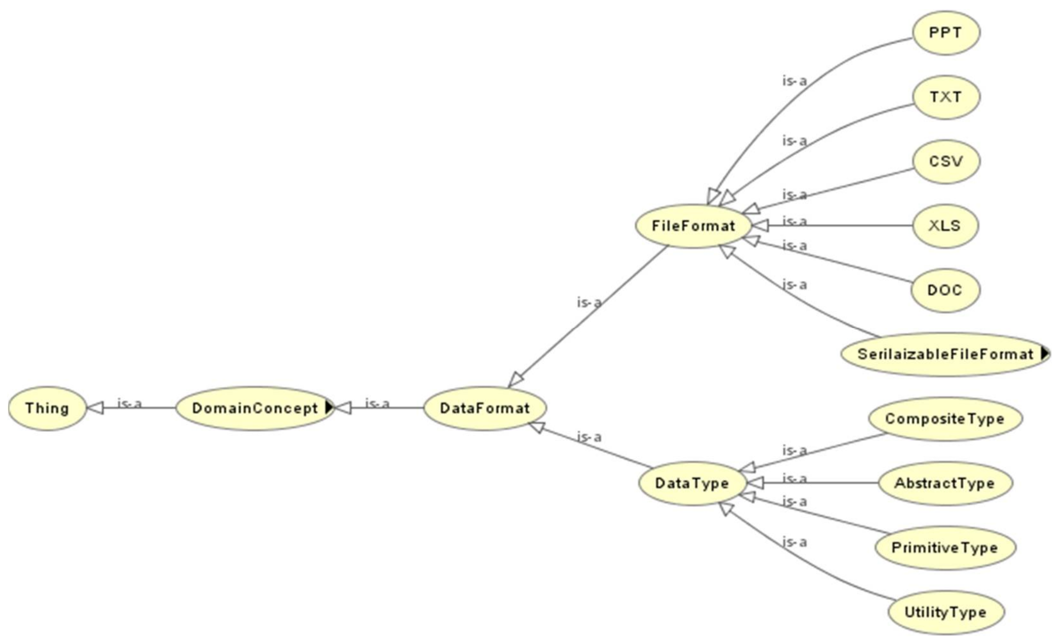

Figure B4: Extract of Data Format Ontology in Taxonomical Form

V. Recipient Ontology

The recipient ontology is simple on the surface. However it becomes more complex when the type of recipients is taken into consideration. These can be organizational entities such as regulators, which are captured in Regulator ontology or humans, captured in the Actor ontology. The recipient ontology is depicted in Figure B5.

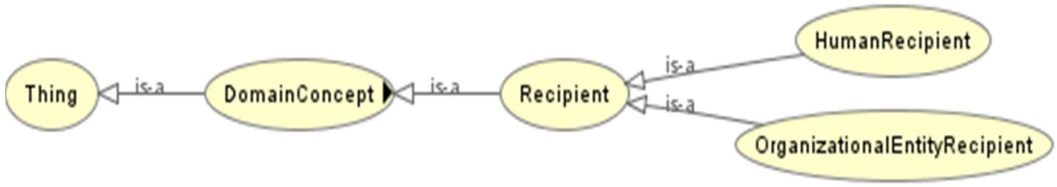

Figure B5: Extract of Recipient Ontology in Taxonomical Form

VI. Regulator Ontology

Regulators are government agencies that oversee a particular economic activity. In the case of the financial services domain, regulators provide an oversight and control function over financial services activities. Regulators have one to many jurisdictions, which refers to the geographic areas that fall under their control. This in captured in the Regulatory Jurisdiction class. A regulatory jurisdiction can be a country, a continent or global.

Regulators can also have control over one to many financial instruments. For example CFTC has oversight functions over certain financial instruments such as derivatives whereas SEC has oversight over equities. Both oversee economic activities which have impact on the US financial services by the activity involving US entities. As such their oversight can have a global reach if the counterparty like a financial services organization is domiciled in a jurisdiction outside the US but is trading with a US entity. In this cases there would be more than one regulator overseeing the activity such as CTFC (US), ESMA (EU), and FCA (UK). There are financial instruments for which oversight is shared between two regulators. For example, there are trading activities on financial instruments that are monitored by both SEC and CFTC.

An extract of the Regulator ontology is depicted in Figure B6. The financial instrument class is represented in third party ontology this appendix. The classes, individuals, properties for the regulator ontology are defined in Table B4.

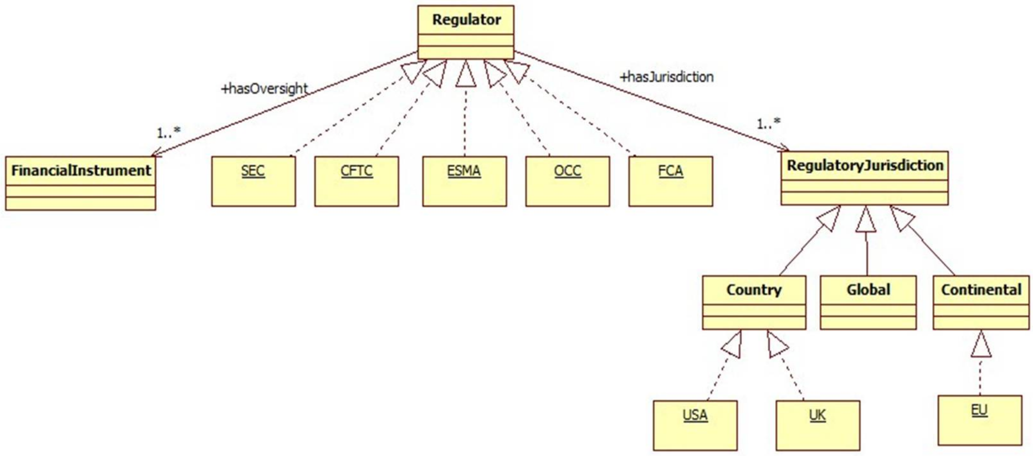

Figure B6: Extract of Regulator Ontology in UML

Table B4: Regulator Ontology Definitions

Term	Definition	Type	Synonym
Regulator	Government entity with oversight function over	Class	
Financial Instrument	Product traded in a financial activity	Class	
has Oversight	Regulators have oversight over trading financial	Property	
SEC	A US Regulator	Individual	
CFTC	A US Regulator	Individual	
ESMA	An EU Regulator	Individual	
OCC	A US Regulator	Individual	
FCA	A US Regulator	Individual	
Regulatory Jurisdiction	Region over which Regulator has oversight	Class	
has Jurisdiction	Retention has a reason	Property	
Country	Distinct entity in political geography	Individual	Nation
USA	USA as a Country	Individual	
UK	UK as a Country	Individual	
Global	Universal geographical jurisdiction	Class	
Continental	Geographical or political grouping of countries	Class	
EU	EU as a continent	Individual	Europe

VII. Actor Ontology

Whereas, a regulator is an organizational entity, an actor is a human. In the context of the REFINTO domain ontology an actor could be a Business Actor or an IT Actor, according to their functions within an organizational entity. An actor has a role within the context of systems within the organizational entity. This is captured in the Actor Role class. An actor can also have access right, captured in the Actor Right class. An extract of the actor ontology is depicted in Figure B7. The classes, individuals and properties are defined in Table B5.

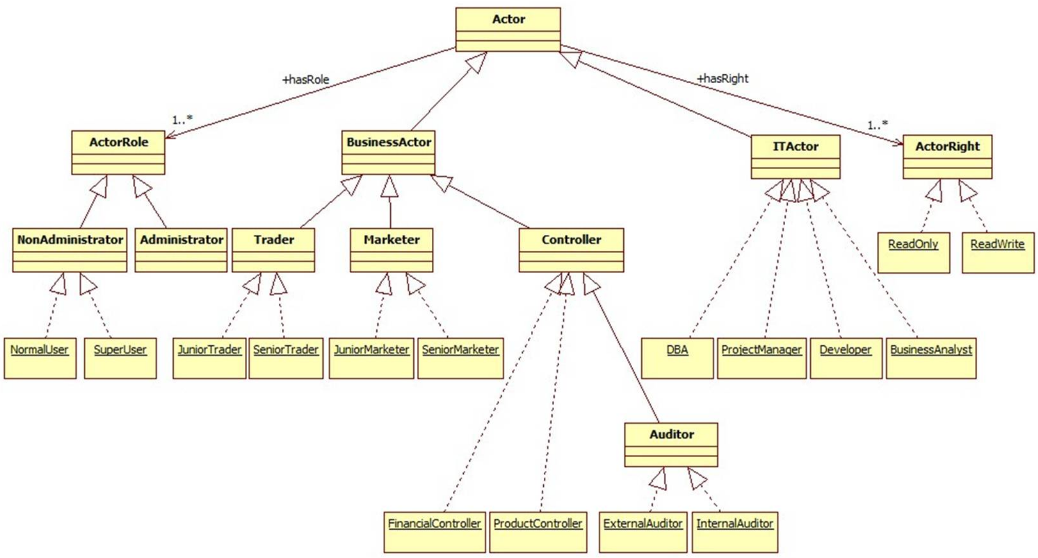

Figure B7: Extract of Actor Ontology in UML

Table B5: Actor Ontology Definitions

Term	Definition	Type	Synonym
Actor	A person who performs some function or receives service with an organizational entity	Class	
Actor Role	he role the actor performs on an organizational entity systems	Class	
has Role	An actor has a role	Property	
Non Administrator	An actor role that restricted from privileged functions	Class	
Normal User	A non-administrator with defined privileges	Individual	
Super User	A non-administrator with defined privileges	Individual	
Administrator	An actor with access to all privileged functions	Individual	
Business Actor	An actor who performs Business functions in the organization	Class	
Marketer	An actor who markets financial instruments counterparties	Class	

Junior Marketer	A marketer in junior cadre	Individual	
Senior Marketer	A marketer in senior cadre	Individual	
Trader	An actor who books trades on financial instruments	Class	
Junior Trader	A trader in junior cadre	Individual	
Senior Trader	A trader in senior cadre	Class	
Controller	An actor who performs oversight functions	Class	
Financial Controller	Business actor who performs financial transactions oversight functions	Individual	
Product Controller	Business actor who performs financial instrument analysis and control oversight functions	Individual	
Auditor	A person with mandate to provide independent review on controls of procedures and activities (finance and IT in this case)	Class	
Internal Auditor	Auditor who is an employee of the organizational entity	Individual	
External Auditor	Auditor who is not an employee of the organizational entity	Individual	
IT Actor	An actor who performs IT functions in the organization	Class	
DBA	IT Actor who performs database management functions	Individual	
Project Manager	IT Actor who performs project governance and control functions	Individual	
Developer	IT Actor who performs application development functions	Individual	
Business Analyst	IT Actor who performs analysis of business needs and analysis and documentation functions	Individual	
Actor Right	Rights that an actor has on process input/outcome	Class	
has Right	An actor has rights	Property	
Read Only	Actor right that allows viewing of process input/outcome	Individual	
Read Write	Actor right that only allows editing of process input/outcome	Individual	

VIII. Process Outcome Ontology

There is a high level of process automation in the financial services domain. These processes are time sensitive and it is common to schedule jobs to run these processes at predefined times. Adherence or non-adherence to these stipulated times has material impact to service level agreements (SLA) that the business entity or sub entities may have with internal and external recipients of the output of the process. Violations of SLAs can lead to financial losses or penalties from regulators.

A given process's output can be an input to other processes. Therefore, there is a high dependency between processes. This makes it imperative that the output of a process be made known to recipients and actors – which may be human or machine (systems). The process outcome ontology is simple and is depicted in Figure B8. A process outcome can run to completion. A completed run outcome can be success, success with exception, or failure. A process run can result in an exception for various reasons. This includes a server being unavailable, required data not available or a timeout.

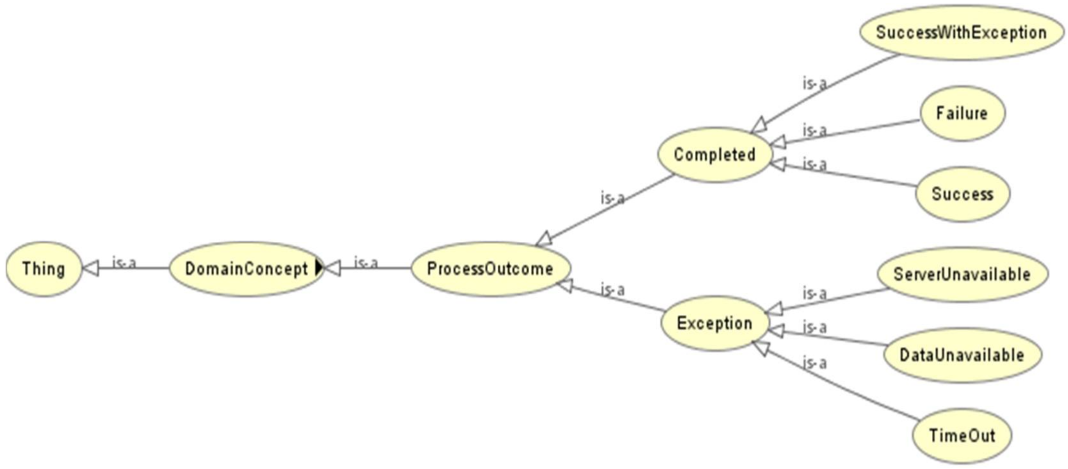

Figure B8: Extract of Process Outcome Ontology in Taxonomical Form

IX. Run Mode Ontology

Processes can be run manually (by humans), automated (machines) or semi-automated (machine with some human intervention). This is captured in the simple Run Mode ontology as depicted in Figure B9.

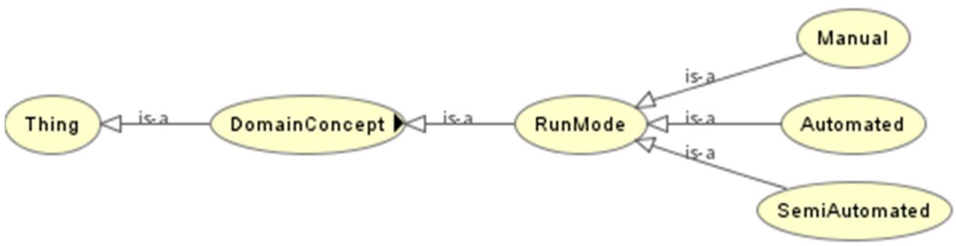

Figure B9: Extract of Run Mode Ontology in Taxonomical Form

X. System Ontology

In the context of the financial services domain, a system collectively refers to software application or collection of software applications and the infrastructure on which it runs to provide a platform that facilitates the performing of a business process or activity. Systems by itself or in collaboration with other systems perform a specific function. It is also not unusual for a system to perform more than one function. The system ontology is depicted in Figure B10. A system can be a host implying that a process may not on it, a destination, which implies it is a target of a process or a source, from which process input can be extracted. Reference data and Trade systems are instances of system which specific functions.

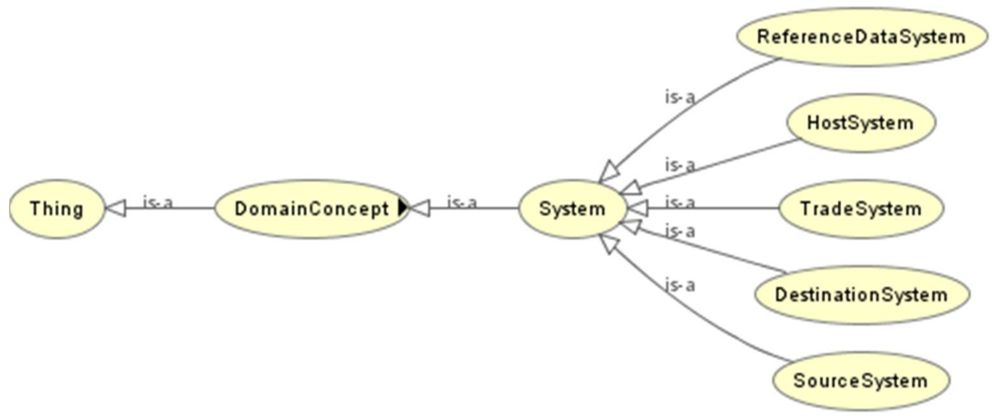

Figure B10: Extract of System Ontology in Taxonomical Form

XI. Reporting Ontology

The reporting ontology is designed to guide requirements elicitation for reporting. Reporting involves providing information on volumes, status, of items of interest within the financial services domain such as trades to interested parties (recipients) that demand the reports such as regulators, auditors, or senior management. Examples of such reports include regulatory reports, management level reports (MIS), and regular reports required by actors such as traders, markers and controllers for various purposes such as risk management, profit and loss calculation, trend analysis, productivity, and efficiency monitoring. The reporting ontology is shown in Figure B11. The classes, individuals and properties are defined in Table B6.

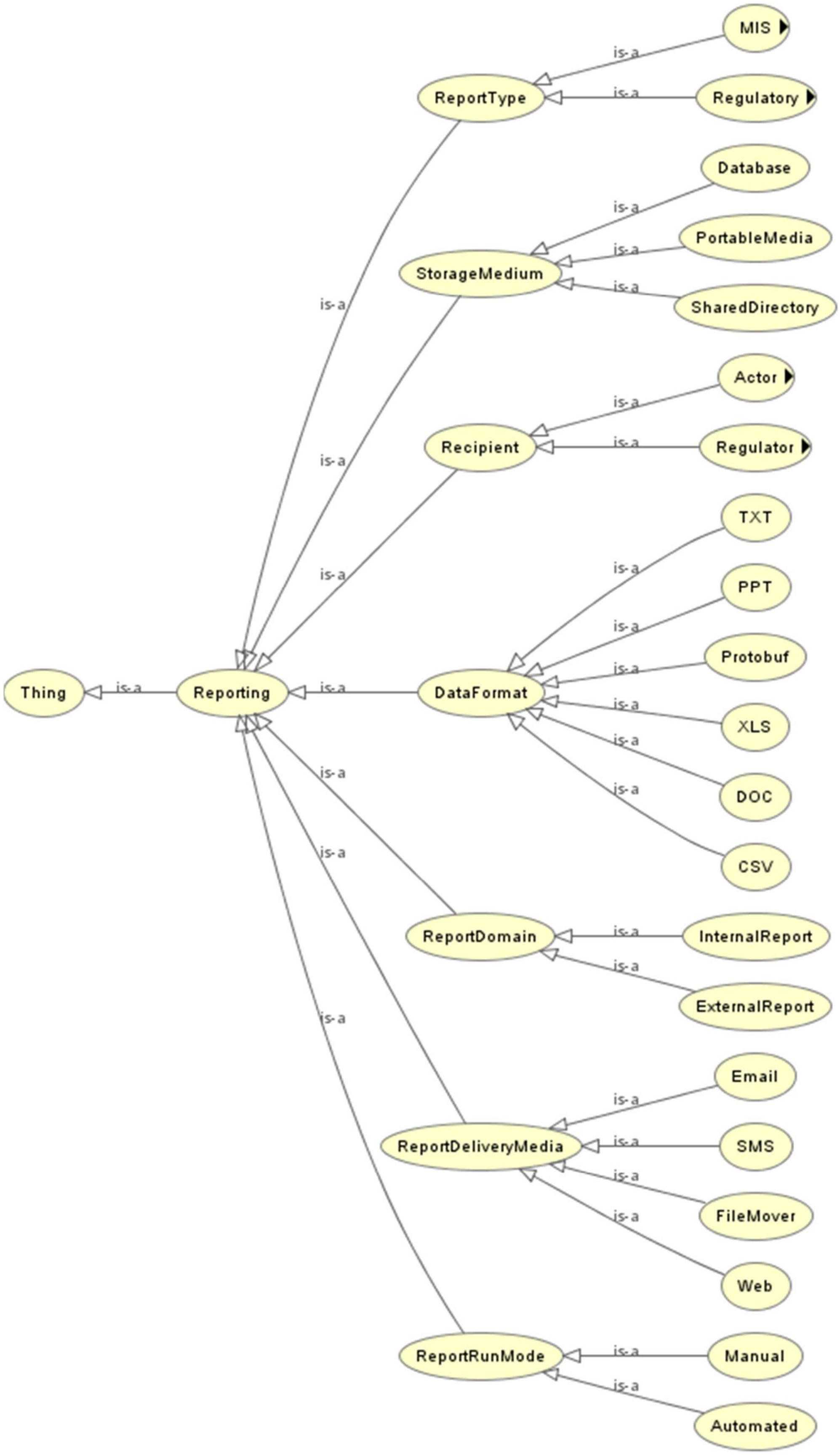

Figure B11: Extract of Reporting Ontology in UML

Table B6: Reporting Ontology Definitions

Term	Definition	Type	Synonym
Report	Amount quoted for a financial instrument	Class	
Run Mode	Refer to Run Mode ontology	Class	
has Run Mode	The type of price that a given price is	Property	
Data Format	Refer to Data Format ontology	Class	
has Data Format	Report has one or many Data Format	Property	
Storage Media	Refer to Storage Media ontology	Class	
has Storage Media	Report may have none or many storage media	Property	
Report Type	The type of Report	Class	
has Report Type	Report have types	Property	
Regulatory	Reports on activities for a Recipient that is a Regulator	Class	
MIS	Management report showing activities and metrics	Individual	
EMIR	Regulatory report for EU derivatives	Individual	
Dodd Frank	Regulatory report for US derivatives	Individual	
Aging Report	MIS report showing ages loans etc.	Individual	
Exception Report	Report showing exceptions in processes or activities	Individual	
Report Domain	The domain where recipients of report function	Class	
has Domain	Report has a domain where Recipient function	Property	
Internal Report	Report meant for internal recipients	Individual	
External Report	Report meant for external recipients	Individual	
Delivery Media	Refer to Delivery Media ontology	Class	
has Delivery Media	Report has means of delivery to Recipient or Storage Media	Individual	
Recipient	Reports may be sent to Recipients	Class	
has Recipient	Report may have none to many recipients	Property	
System	Refer to System ontology	Class	
runs On	Report runs on a System	Property	

XII. Data Source Ontology

Information is the fuel that powers the financial service industry and it is sourced from processing data. The quality of data, the speed at which is obtained and processed and the utilization of the processed data is a distinguishing factor for gaining and maintaining market share. This has both direct and indirect impact on the profitability of the enterprise. Data sourcing is therefore one of the key activities and most frequent reoccurring business requests from the business to IT. Incorrect or untimely data has severe impact on the ability to make risk management decisions and therefore affects the ability of the trading units to make sound judgement or execute trades. Furthermore, it can result in regulator censure and litigation from clients for material misrepresentation. This can result in reputational damage and significant financial losses. An extract of the data ontology is shown in Figure B12.

Figure B12: Extract of Data Souring Ontology in Taxonomical Form

It uses common ontologies like delivery media, retention and data format discussed earlier. The ontology captures and makes explicit knowledge about data sourcing constructs. It can be used to elicit more complete and high quality requirements for this very important activity.

The classes, individuals and properties are defined in Table B7. There are relationships not depicted in the diagram. For example, a source entity usually provides data for specific purpose. External entities like Bloomberg more often than not provide market data.

Table B7: Data Souring Ontology Definitions

Term	Definition	Type	Synonym
Data	Details about phenomena sourced an from entity presented in usable form appropriate for subscriber needs	Class	
Delivery Media	Refer to Delivery Media ontology	Class	
has Delivery Media	Data has Delivery Media	Property	
Data Format	Refer to Data Format ontology	Class	
has Format	Data has format/type	Property	
Refresh Frequency	Frequency at which data is refreshed	Class	
Refresh Real time	Instance of Refresh done immediately an event occurs	Individual	
Hourly Refresh	Instance of Refresh done hourly	Individual	
Daily Refresh	Instance of Refresh done daily	Individual	
Weekly Refresh	Instance of Refresh done weekly	Individual	
Monthly Refresh	Instance of Refresh done monthly	Individual	
Yearly Refresh	Instance of Refresh done yearly	Individual	
Data Purpose	Purpose for which data is sourced for	Class	
has Purpose	Data has purpose	Property	
Transaction Data	Data generated through activities of an entity	Class	
Trade Data	Instance of Transaction data for trading activities	Individual	
Market Data	Publicly data usually for a subscription fee/license	Class	
Corporate Action Data	Instance of Market Data for activities like Dividends	Individual	
Reference Data	Internally generated data used as reference for activities	Class	
Counterparty Data	Instance of Reference Data on clients	Individual	
Source Entity	Entity that collates and distribute Data	Class	
is Source From	Data is sourced from a Source Entity	Property	
External Entity	Source Entity that is outside consuming entity	Class	
Internal Entity	Source Entity that is within consuming entity	Class	
Markit	Instance of External Entity	Individual	
Bloomberg	Instance of External Entity	Individual	

Reuters	Instance of Internal Entity	Individual	
Risk Analytics	Instance of Internal Entity	Individual	
Treasury	Instance of Internal Entity	Individual	
Retention	Refer to Retention ontology	Class	
Is Retained For	Data is retained for specified retention period	Property	

XIII. Reconciliation Ontology

Reconciliation is one of the most important and common used tools for activity governance and controls in the financial services domain. Running reconciliations on a plethora of activities at scheduled times or real time is a key function of middle offices. In its most basic form, it verification of details typically held in one or more systems (refer to System ontology) against those held on other systems. The system can be based within the entity or externally. The reconciliation ontology depicted in Figure B13 and uses the common ontology that has been discussed already. The classes, individuals and properties are defined in Table B8.

Table B8: Reconciliation Ontology Definitions

Term	Definition	Type	Synonym
Reconciliation	Cross check of details on concept on different systems on defined criteria and logic like accuracy, completeness etc.	Class	
Data	Refer to Data ontology	Class	
is Based On	Reconciliation is based on Data	Property	
System	Refer to System ontology	Class	
is Run On	Reconciliation runs on Systems	Property	
Process Outcome	Refer to Process Outcome ontology	Class	
has Outcome	Reconciliation run process has outcome	Property	
Recipient	Refer to Recipient ontology	Class	
has Recipient	Reconciliation has a Recipient	Property	
Storage Media	Refer to Storage Media ontology	Class	
is Stored In	Reconciliation output is stored in Storage Media	Property	
Delivery Media	Refer to Delivery Media ontology	Class	
has Delivery Media	Reconciliation output has Delivery Media to Recipients	Property	
Run Mode	Refer to Run Mode ontology	Class	
has Run Mode	Reconciliation has Run Mode	Property	

Figure B13: Extract of Reconciliation Ontology in Taxonomical Form

XIV. Control Ontology

The control ontology is depicted in Figure B14. Activities and processes in the financial services domain require constant monitoring. It is common for organizational policy or regulatory stipulation to state acceptable thresholds for various measures. Within the organizational entity there are SLAs between the business and IT that needs to be continuously measure to ensure that the business is running optimally and when necessary take predefined action based on criteria, threshold and measurement units.

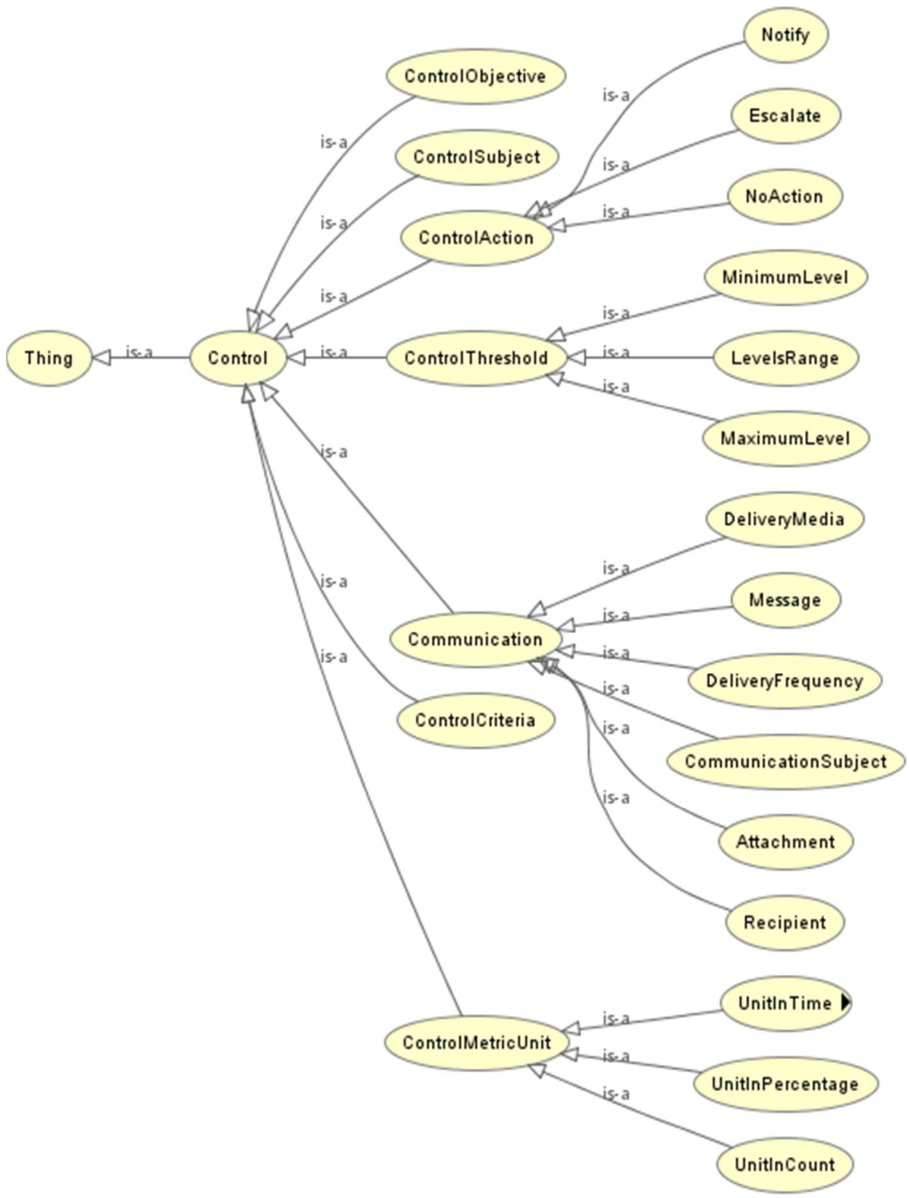

Figure B14: Extract of Control Ontology

Controls provide an early warning when anomalies occur and can be very useful in giving the actors responsible for taking appropriate remediation action at earliest opportunity to avoid knock on effects on other dependent activities and process. A timely notification of control violations is essential for proactively containing financial losses and avoidance of reputational damage.

The control ontology captures knowledge about this very critical activity in the domain and can be used in the requirements elicitation process for control needs. The control ontology uses common ontologies discussed earlier such as Delivery Media, System and Run Mode. The definition of the classes, individuals and properties is defined in Table B9

Table B9: Control Ontology Definitions

Term	Definition	Type	Synonym
Control	A process to monitor activities in a domain	Class	
Control Criteria	Criteria by which a Control is observed or measured by	Class	
has Criteria	Control has criteria	Property	
Control Threshold	Bounds control is measured by	Class	
Minimum Level	Lower bound for control criteria	Individual	
Range Of Level	Level between Lower and Upper bounds	Individual	
Maximum Level	Upper bound for control criteria	Individual	
Control Metric Unit	Unit by which control criteria is measured by	Class	
Control Unit In Time	Control criteria measured by time	Class	
Control Unit In Count	Control criteria measured by count	Property	
Control Unit In Percentage	Control criteria measured by percentage points	Individual	
Control Unit In Minutes	Instance of Time Control unit measurement in Minutes	Individual	
Control Unit In Hours	Instance of Time Control unit measurement in Hours	Individual	
Control Unit In Days	Instance of Time Control unit measurement in Days	Individual	
Control Unit In Months	Instance of Time Control unit measurement in Months	Individual	
Control Unit In Years	Instance of Time Control unit measurement in Years	Individual	
Control Objective	Purpose of the Control	Class	
has Objective	Control has objective(s)	Property	
Control Action	Action to be taken when Control Criteria reach Threshold	Class	
has Control Action	Control has Action to be taken if defined Threshold is reached	Property	

Notify	Instance of control Action to inform interested Recipient(s)	Individual	
No Action	Instance of control Action to do nothing	Individual	
Escalate	Instance of control Action to inform other Recipient(s) if further Action not taken by original Recipient(s)	Individual	
Delivery Media	Refer to Delivery Media Ontology	Class	
has Delivery Media	Control have Delivery Media for Action for example	Property	
System	Refer to System Ontology	Class	
is Run On	Control is run on System	Property	
Run Mode	Refer to Run Mode Ontology	Class	
has Run Mode	Control has a Run Mode	Property	

XV. Pricing Ontology

Pricing is one of the cardinal activities in financial services. It is a complex process that takes various inputs and produces outputs one of which is the price in the desired currency and the time period that the price is valid for. The price ontology is a depiction of the terms and concepts that capture knowledge of pricing. The ontology uses common ontology like Delivery Media and Recipient ontology already discussed. The main classes in the pricing ontology include Price Type, Recipient, Parameter, and Financial Instrument.

The Financial Instrument Business ontology (Vieira, 2011) provides a rich semantics that can be used to actualize the Financial Instrument class of the REFINTO framework pricing ontology. The Financial Instrument ontology from the EDM council addresses in considerable detail, the family of asset classes such as debt, equities, derivatives etc. There was therefore no need to redesign ontology for financial instruments.

An extract of the pricing ontology is shown in Figure B15. The classes, individuals, properties and multiplicities between the classes are defined in Table B10.

Figure B15: Extract of Pricing Ontology

Table B10: Pricing Ontology Definitions

Term	Definition	Type	Synonym
Price	Government entity with oversight function over financial activities	Class	Quote
Price Type	The type of price	Class	
has Type	The type of price that a given price is	Property	
Indicative	A reasonable estimate of the price often when quantity not specified	Individual	
Firm	Guaranteed price for a financial instrument	Individual	
Parameter	Variables that are required for pricing or output from pricing	Class	
has Parameter	Price has Parameter(s)	Property	
Output Parameter	Output from pricing or attribute of Price	Class	

Input Parameter	Variables that are required for pricing	Class	
Trade Date	Business day that a trade event will happen	Individual	
Settlement Date	Business day that a trade settles, for example T+2 (two days after Trade Date)	Individual	
Expiry Date	The last day that an Options or Futures (Derivatives) contract is valid	Individual	
Financial Instrument	Easily tradable packages of capital each having its own unique characteristics and structure	Class	
Debt	Geographical or political grouping of countries	Class	
Equities	A Financial Instrument that gives the buyer (Investor) a percentage of owner of a Company	Class	Securities
Derivatives	A Financial Instrument that derives its value from another Financial Instrument	Class	
Delivery Media	The media through which a price is conveyed to Recipient	Class	
has Delivery Media	Price has a media through which it is conveyed to the Recipient	Property	
Web	Price can be by published via the web example a web page	Individual	
Email	Price can be conveyed in an email	Individual	
Telephone	Price can be conveyed over the telephone	Individual	
Recipient	A trading entity (Counterparty) that receives a price	Class	
has Recipient	Price may have none to many recipients	Property	
External Recipient	A trading entity (Counterparty) that is outside the who receives a price	Class	
Internal Recipient	A trading entity (Counterparty) that is within the originating entity that offers the price	Class	
Spot Price	Current price a Financial Instrument can be sold at specified time and place	Individual	Current
Strike Price	Price at which a specific derivative contract can be exercised. Used mostly to describe stock and index options in which prices are fixed	Individual	
Bid Price	Price quoted by/to an entity to Sell a Financial Instrument	Individual	Offer
Ask Price	Price quoted by/to an entity to Buy a Financial Instrument	Individual	
Quantity	The number of Financial Instruments on offer for sell	Individual	

XVI. Third-Party Ontologies

There is a rich set of third party ontologies for financial services that can be leveraged in framework such as the REFINTO framework. These include the Financial Industry Business Ontology (FIBO) (EDM Council, 2009), a set of ontologies which cover currencies, markets, parties, financial instruments and other reference data widely and commonly used in the financial industry developed by Fadyart (2009) and DIP eBanking financial ontology (Agarwal et al., 2010). The FIBO ontology covers financial instrument and has been discussed in this appendix. The FIBO ontology is depicted in Figure B16.

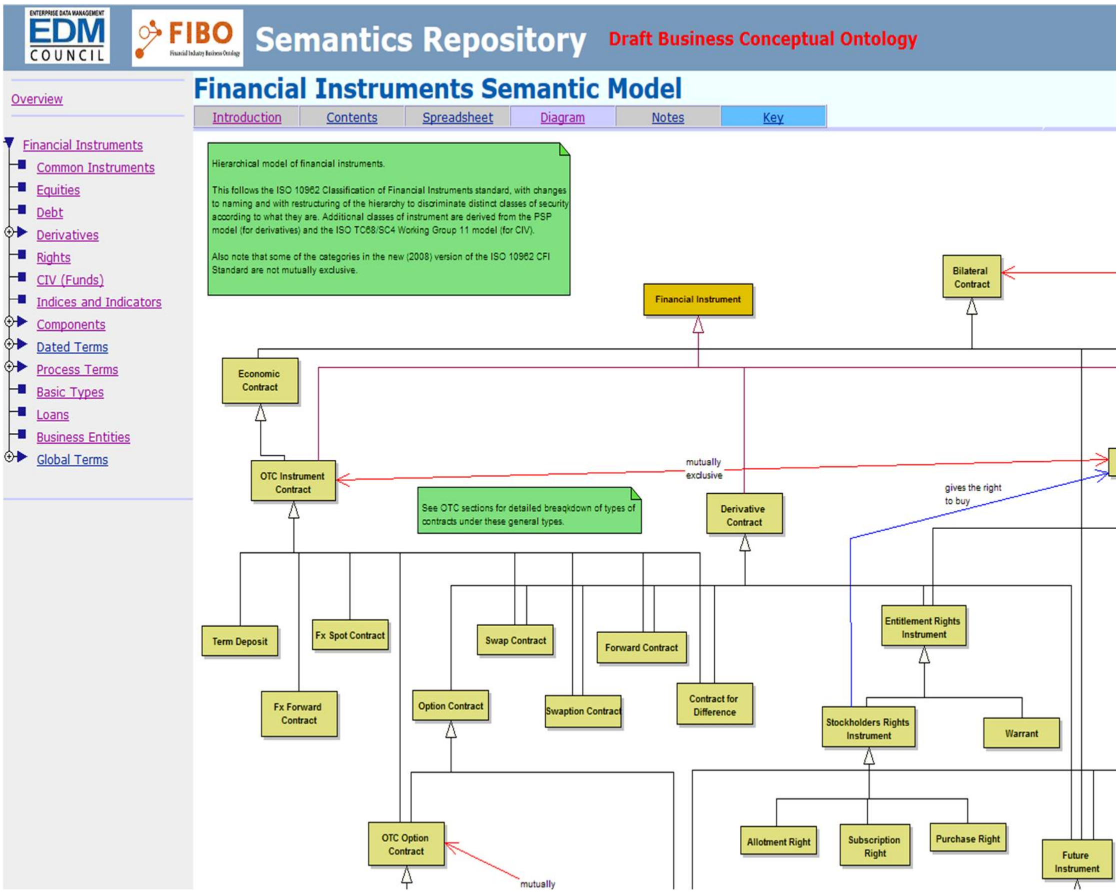

Figure B16: FIBO Ontology (EDM Council, 2009)

The Fadyart finance ontology shown on Figure B17 covers a wider range of concepts including financial instruments, counterparties (parties), currencies, activities etc. There are a number of other third party ontologies but these ones were of particular interest and relevance.

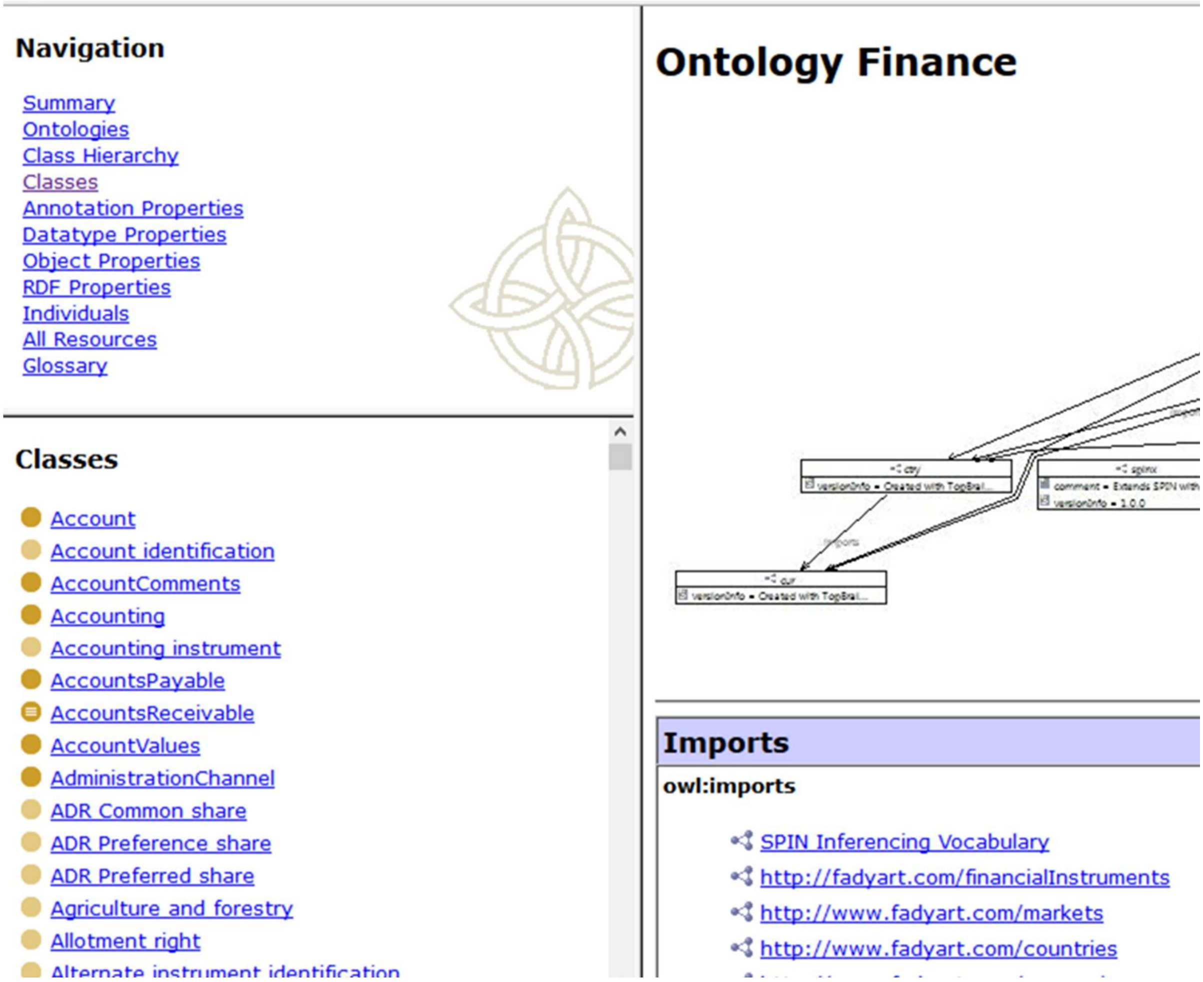

Figure B17: Extract of Currency Ontology (2010)

Appendix C: Research Instrument (Questionnaire)

Item	Question	Hypothesis	Group
Q1	The respondent rates the size of the requirements document is appropriate for the project	H1	Quality
Q2	The respondent found words like ‘will’, ‘shall’, ‘must’ to define deliverables in requirements document	H1	Quality
Q3	The respondent found phrases like ‘the following’ after ‘will’, ‘shall’, and ‘must’ to further detail deliverables	H1	Quality
Q4	The respondent found appropriate contain diagrams, screen shots, links to aid clarity in requirement document	H1	Quality
Q5	Does the requirement document contain words like large, fast, enough, which hint of ambiguity	H1	Quality
Q6	The respondent found acronyms like ‘TBD’ and ‘TBS’ implying incomplete requirements	H1	Quality
Q7	The respondent found words like ‘can’, ‘may’, ‘I/we think’ which hint of uncertainty	H1	Quality
Q8	The respondent opinion on frequency of requirements changes - addition, deletion, and modification	H1	Quality
Q9	The respondent opinion on the structure of requirement document with sections and content positioned appropriately	H1	Quality
Q10	The respondent’s opinion on oversight and control of revisions to requirements document	H1	Quality
Q11	The respondent’s opinion on the quality of the requirements document as a whole	H1	Quality
Q12	The respondent’s opinion on the quality of the artefact that is a result of the requirements	H1	Quality
U1	The respondent’s opinion on the ease of understanding the expression of concepts and vocabularies in the requirements	H2	Common Language
U2	The respondent’s opinion on IT’s understanding of business’s requirement	H2	Common Language
C1	The respondent’s opinion on the interaction and updates on progress, issues and challenges during the project initiation	H3	Common Language
C2	The respondent’s opinion on the interaction and updates on progress, issues and challenges during the project implementation	H3	Common Language

C3	The respondent's opinion on the interaction and updates on progress, issues and challenges after go-live	H3	Common Language
C4	The respondent's opinion the frequency of communication	H3	Common Language
C5	The respondent's opinion the relevance of communication	H3	Common Language
L1	The respondent learnt about the business or technical area during the requirement elicitation stage	H4	Common Language
L2	The respondent learnt about the business processes of the project or technical processes involved in developing solutions for the project	H4	Common Language
L3	The respondent had a better understanding about the business or technical processes at the end of the project	H4	Common Language
R1	The respondent's opinion on the usefulness/relevance of past requirement on the current requirement	H5	Service Orientation
R2	The respondent's opinion on the ease of identifying assets to similar to current requirements	H5	Service Orientation
R3	The respondent's opinion on the accuracy of the documentation on the artefact	H5	Service Orientation
R4	The respondent's opinion on the relevance of the assets identified for reuse was in context of current requirements	H5	Service Orientation
R5	The respondent's opinion on the overall effect of using of past requirement to clarify and refine current requirement	H5	Service Orientation
E1	The respondent's opinion on the frequency of scrapped or reworked parts of the project	H6	Continuous Monitoring
E2	The respondent's opinion of the lessons learnt review process	H6	Continuous Monitoring
F1	The respondent's rating of the project in terms of estimated and actual project staffing	H7	Continuous Monitoring
F2	The respondent's rating of project delivery on milestones against project plan	H7	Continuous Monitoring
S1	The respondent's opinion on resource efficiency due to the requirement elicitation process adopted	H8	Continuous Monitoring
S2	The respondent's opinion on resource efficiency due to the method of matching and reuse adopted	H8	Continuous Monitoring
S3	The respondent's opinion on resource efficiency due to the framework adopted for project	H8	Continuous Monitoring

Appendix D: Data Collection

Data Collection for NON-REFINTO Portfolio

Data collection for projects executed with ad hoc methods (NON-REFINTO) was carried out after each of the three stages in the project lifecycle. There is no formalized structure for requirements elicitation in this category of projects. Project participants were given the same questionnaire as those following the REFINTO framework for the pre-project, intra-project and post project stages with explanations on areas which are not relevant such as matching artefacts with requirements in intra project stage which may follow processes different from the REFINTO framework process at this stage.

The breakdown of participants within each project which corresponds to the number of respondents for the surveys is provided in Table 0.1. The NON-REFINTO project portfolio breakdown is shown in Table 0.2.

Table 0.1: Roles of NON-REFINTO Participants

Role	Observed N
Sponsor	2
Subject Matter Expert	3
Business User	35
Project Manager	6
Business Analyst	4
Requirements Engineer	1
Developer	5
Total	56

Table 0.2: NON-REFINTO Portfolio Breakdown

Classification	Count
Control	3
Reporting	4
Data Sourcing	1
Reconciliation	2

Data Collection for RFINTO-ONLY Portfolio

Data collection was performed after each stage of projects in this portfolio. Although there is no tool support for function such as semi-automated requirements elicitation and requirements to arterfact matching, the questionnaire measured equivalent processes to automation functions tool support provides. The breakdown of participants for the portfolio which corresponds to the number of respondents for the surveys is provided in Table 0.3.

Table 0.3: Roles of REFINTO ONLY Participants

Role	Observed N
Sponsor	1
Subject Matter Expert	4
Business User	39
Project Manager	4
Business Analyst	5
Requirements Engineer	2
Developer	6
Total	61

The REFINTO ONLY project portfolio breakdown is shown in Table 0.4.

Table 0.4: REFINTO ONLY Portfolio Breakdown

Classification	Count
Reconciliation	2
Control	4
Regulatory	1
Reporting	2
Data Sourcing	1

Data Collection for REFINTO and FRAMEWORK Portfolio

Data collection was performed after each stage of projects in this portfolio. Tool support is used for functions such as semi-automated requirements elicitation and requirements to arterfact matching. The breakdown of participants within each project which corresponds to the number of respondents for the surveys is provided in Table 0.5. The REFINTO and TOOL portfolio breakdown is shown in Table 0.6.

Table 0.5: Roles of REFINTO and TOOLS Participants

Role	Observed N
Sponsor	2
Subject Matter Expert	3
Business User	38
Project Manager	4
Business Analyst	4
Requirements Engineer	2
Developer	5
Total	58

Table 0.6: REFINTO and TOOL Portfolio Breakdown

Classification	Description
Control	3
Reporting	3
Data Sourcing	1
Regulatory	1
Reconciliation	2

Data Collection for THIRD-PARTY Portfolio

For projects executed with third party processes, the standard data collection processes followed for the other three portfolios were used. Equivalent points to the three projects stages were identified and participants were given the same questionnaire as the other project portfolios with explanations provided for the contexts of questions particularly for the intra-project stage. The breakdown of participants within each project which corresponds to the number of respondents for the surveys is provided in Table 0.7. The THIRD PARTY project portfolio breakdown is shown in Table 0.8.

Table 0.7: Roles of THIRD PARTY Participants

Role	Observed N
Sponsor	1
Subject Matter Expert	3
Business User	33
Project Manager	3
Business Analyst	4
Requirements Engineer	1
Developer	7
Total	52

Table 0.8: THIRD PARTY Portfolio Breakdown

Classification	Count
Reporting	2
Reconciliation	4
Control	4

Data Collection by Stage

In this section, the breakdown of participants responding to each project stage for all the projects in all four portfolios is provided.

Data Collection at Pre-Project Stage

The data collection rate attained in at the pre-project stage for all the projects is an average of 92.1%. In Table 0.9, the breakdown of participants in NON-REFINTO project portfolios at the pre-project stage shows a response rate of 91.1% providing a total of 51 samples.

Table 0.9: NON-REFINTO Participants at Pre-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	2	-1
Subject Matter Expert	3	3	0
Business User	32	35	-3
Project Manager	5	6	-1
Business Analyst	4	4	0
Requirements Engineer	1	1	0
Developer	5	5	0
Total	51	56	

In Table 0.10, the breakdown for REFINTO ONLY projects at the pre-project stage shows a response rate of 93.4% providing a total of 57 samples.

Table 0.10: REFINTO ONLY Participants at Pre-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	1	0
Subject Matter Expert	2	4	-2
Business User	37	39	-2
Project Manager	4	4	0
Business Analyst	5	5	0
Requirements Engineer	2	2	0
Developer	6	6	0
Total	57	61	

In Table 0.11, the breakdown for REFINTO and TOOLS projects at the pre-project stage shows a response rate of 91.4% providing a total of 53 samples.

Table 0.11: REFINTO and TOOLS Participants at Pre-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	2	2	0
Subject Matter Expert	2	3	-1
Business User	36	38	-2
Project Manager	3	4	-1
Business Analyst	4	4	0
Requirements Engineer	2	2	0
Developer	4	5	-1
Total	53	58	

In Table 0.12, the breakdown for THIRD PARTY projects at the pre-project stage shows a response rate of 92.3% providing a total of 48 samples.

Table 0.12: THIRD PARTY Participants at Pre-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	1	0
Subject Matter Expert	2	3	-1
Business User	32	33	-1
Project Manager	3	3	0
Business Analyst	3	4	-1
Requirements Engineer	1	1	0
Developer	6	7	-1
Total	48	52	

The total number of data samples collected for the pre-project stage is 209. The samples accepted for analysis and considered useful is 100%.

Data Collection at Intra-Project Stage

The data collection rate attained in at the pre-project stage for all the projects is 93.8%. In Table 0.13, the breakdown of participants NON-REFINTO projects at the pre-project stage shows a response rate of 94.6% providing a total of 53 samples.

Table 0.13: NON-REFINTO Participants at Intra-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	2	-1
Subject Matter Expert	2	3	-1
Business User	34	35	-1
Project Manager	6	6	0
Business Analyst	4	4	0
Requirements Engineer	1	1	0
Developer	5	5	0
Total	53	56	

In Table 0.14, the breakdown for REFINTO ONLY projects at the pre-project stage shows a response rate of 93.4% providing a total of 57 samples

Table 0.14: REFINTO ONLY Participants at Intra-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	1	0
Subject Matter Expert	3	4	-1
Business User	37	39	-2
Project Manager	4	4	0
Business Analyst	5	5	0
Requirements Engineer	2	2	0
Developer	5	6	-1
Total	57	61	

In Table 0.15, the breakdown for REFINTO and TOOLS projects at the pre-project stage shows a response rate of 93.1% providing a total of 54 samples.

Table 0.15: REFINTO and TOOLS Participants at Intra-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	2	2	0
Subject Matter Expert	3	3	0
Business User	36	38	-2
Project Manager	4	4	0
Business Analyst	3	4	-1
Requirements Engineer	1	2	-1
Developer	5	5	0
Total	54	58	

In Table 0.16, the breakdown for THIRD PARTY projects at the pre-project stage shows a response rate of 94.2% providing a total of 49 samples.

Table 0.16: THIRD PARTY Participants at Intra-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	1	0
Subject Matter Expert	3	3	0
Business User	31	33	-2
Project Manager	3	3	0
Business Analyst	4	4	0
Requirements Engineer	1	1	0
Developer	6	7	-1
Total	49	52	

The total number of data samples collected for the pre-project stage is 213. The samples accepted for analysis and considered useful is 100%.

Data Collection at Post-Project Stage

The data collection rate attained in at the pre-project stage for all the projects is about 93.4%. In Table 0.17, the breakdown of participants in NON-REFINTO projects portfolio at the pre-project stage shows a response rate of 92.8% providing a total of 52 samples.

Table 0.17: NON-REFINTO Participants at Post-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	2	2	0
Subject Matter Expert	3	3	0
Business User	32	35	-3
Project Manager	5	4	1
Business Analyst	4	4	0
Requirements Engineer	1	1	0
Developer	5	5	0
Total	52	56	

In Table 0.18, the breakdown for REFINTO ONLY projects at the pre-project stage shows a response rate of 93.4% providing a total of 57 samples.

Table 0.18: REFINTO ONLY Participants at Post-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	1	0
Subject Matter Expert	4	4	0
Business User	37	39	-2
Project Manager	3	4	-1
Business Analyst	4	5	-1
Requirements Engineer	2	2	0
Developer	6	6	0
Total	57	61	

In Table 0.19, the breakdown for REFINTO and TOOLS projects at the pre-project stage shows a response rate of 93.1% providing a total of 54 samples.

Table 0.19: REFINTO ONLY Participants at Post-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	2	2	0
Subject Matter Expert	2	3	-1
Business User	36	38	-2
Project Manager	4	4	0
Business Analyst	4	4	0
Requirements Engineer	2	2	0
Developer	4	5	-1
Total	54	58	

In Table 0.20, the breakdown for THIRD PARTY projects at the pre-project stage shows a response rate of 94.2% providing a total of 49 samples.

Table 0.20: THIRD PARTY Participants at Post-Project Stage

Role	Observed N	Expected N	Residual
Sponsor	1	1	0
Subject Matter Expert	3	3	0
Business User	32	33	-1
Project Manager	3	3	0
Business Analyst	1	3	-2
Requirements Engineer	1	1	0
Developer	7	7	0
Total	49	52	

The total number of data samples collected for the pre-project stage is 212. The percentage of samples accepted for analysis and considered useful is 100%.