Collaborative Requirements Elicitation

To Foster

Shared Understanding

A Thesis submitted to the

University of Dublin, Trinity College

For the degree of

Doctor in Philosophy

Anthony Paul Stynes

Knowledge and Data Engineering Group,

School of Computer Science and Statistics,

Trinity College Dublin,

Submitted April 2015

Declaration

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

_____

Anthony Paul Stynes

April 2015

Permission to lend or copy

I, the undersigned, agree that the Trinity College Library may lend or copy this thesis upon request.

_____

Anthony Paul Stynes

April 2015

ACKNOWLEGDEMENTS

Many people have influenced the writing of this thesis and provided both guidance and support on what has been a challenging, yet enjoyable and rewarding journey. I would like to acknowledge these individuals and thank them for their contributions that they have made over the past number of years which have made this thesis and the work described herein possible.

Firstly I would like to thank my supervisors Professor Declan O'Sullivan and Professor Owen Conlan for all their support, encouragement and advice throughout this research.

I would like to especially thank my parents Anthony and Breda whose constant support, help and encouragement made it possible for me to complete this work.

Thanks also to my children Thomas, Aoibheann and Joseph who brought great joy and helped to keep a balance between work and family.

ABSTRACT

Fostering shared understanding of how to support organisational change involves a complex interaction of stakeholders in order to achieve alignment of business processes and IT infrastructure. Collaboration and communication among stakeholders will determine the success of any organisational change initiative. The quality of alignment between the business processes and IT infrastructure will determine the business value of that organisational change. Achieving a strong alignment requires a combination of appropriate requirement elicitation techniques to gain an understanding of the business requirements. New approaches to collaborative requirements elicitation that fosters shared understanding amongst stakeholders need to address the issue of IT alignment.

The contribution of this thesis to the area is an innovative collaborative requirements elicitation framework that introduces a novel combination of collaboration, communication, prototyping and modelling techniques in a manner that differs significantly from current collaborative requirement elicitation frameworks.

This thesis presents the design of the theoretical framework that comprises of a process model with four activities. The thesis also describes the design of a support tool that implements the two central activities of the process model. In order to determine the extent shared understanding was fostered, the framework was evaluated in a case study of a business process for the roll out of the IT software image at a third level educational institution. The evaluation involved two experiments by stakeholders that elicited requirements for aligning the business process of the institution with an IT infrastructure. Statistical analysis showed that the collaborative requirement elicitation framework fostered shared understanding in the case study, through increases in communication and increases in stakeholder's positive experience with collaboration. Shared understanding is also manifested in the creation of three knowledge representation artefacts namely, requirements model, IT infrastructure model, and a business process model. Statistical analysis also showed that the IT infrastructure model has the potential to represent a realistic solution.

The framework will be useful to requirements engineers and business analysts that work on designing and implementing business processes. It provides them with a set of instructions and a support tool that will allow them to collaborate and communicate business requirements in a semantically consistent and understandable manner, and then reflect the potential impact of those requirements on the alignment of the business process and IT infrastructure.

TABLE OF CONTENTS

6

TABLE OF FIGURES

TABLES

ABBREVIATIONS

ACE   Attempto Controlled English

APE   ACE Parsing Engine

BPEL   Business Process Execution Language

BPMN   Business Process Modelling and Notation

CRESUS   Collaborative Requirements Elicitation through Enhancing Shared Understanding and Scenarios

NCI   National College of Ireland

CRESUS-T   Collaborative Requirements Elicitation through Enhancing Shared Understanding and Scenarios – Tool

GUI   Graphic User Interface

IOT   Institute of Technology

JDOM   Java-based Document Object Model

NFQ   National Framework of Qualifications

OWL   Web Ontology Language

RDF   Resource Description Framework

TrAX   Transformation API for XML

XML   Extensible Markup Language

XPATH   XML Path Language

XQuery   XML Query Language

XSL   Extensible Stylesheet Language

XSLT   Extensible Stylesheet Language Transformations

1 Introduction

1.1 Motivation

One of the great moral philosophers Adam Smith (1776) describes the division of labour in representing an early example of a business process. Smith (1776) describes the production of a pin in his famous example of a pin factory in the following way.

> "One man draws out the wire; another straights it; a third cuts it; a fourth points it; a fifth grinds it at the top for receiving the head; to make the head requires two or three distinct operations; to put it on is a particular business; to whiten the pins is another; it is even a trade by itself to put them into the paper; and the important business of making a pin is, in this manner, divided into about eighteen distinct operations, which, in some manufactories, are all performed by distinct hands, though in others the same man will sometimes perform two or three of them."

Smith (1776) describes the business process as a set of simple activities performed by specialised workers. Much has changed since. Business processes are still a set of structured activities that produce a specific service or product. However, IT infrastructure tends to be a key component that automates a lot of the human activity, as for example IT Incident Management business process (Bulles et al., 2010). Modern approaches to creating business processes tend to involve a complex interaction of stakeholders, in order to achieve alignment[1] of business processes and IT infrastructure.

Luftman, (2000) evaluates organisations' business-IT alignment in terms of where they are and what they can do to improve alignment. Weak alignment is characterised by no formal processes in which business executives and IT Architects lack understanding; don't liaise or liaise as needed; the role of IT in strategic business planning is not

---

[1] IT and business alignment has consistently been a key issue for business executives (Luftman et al., 2006; Luftmann & Kempaiah, 2008; Luftman & Ben-Zvi, 2010a; Luftman & Ben-Zvi, 2010b; Luftman & Ben-Zvi, 2011; Luftman & Derksen, 2012). According to Reich & Benbasat (2000) there are two dimensions to alignment. The intellectual dimension of alignment is defined as "the state in which a high-quality set of interrelated IT and business plans exists." The social dimension of alignment is defined as "the state in which business and IT executives within an organizational unit understand and are committed to the business and IT mission, objectives, and plans" (Reich and Benbasat 1996).

involved; IT infrastructure is not well integrated and is perceived as a utility run at minimum cost. Strong alignment is characterised by optimised processes where both business executives and IT architects have a shared understanding of each other's processes; business executives and IT architects communicate knowledge and collaborate with each other; the role of IT in strategic business planning drives business processes and adapts quickly to change; IT infrastructure is integrated across functions and is perceived as being able to respond fast to changing markets.

Currently, the majority of software is not developed in house but is available for purchase or for free, such as customer resource management (CRM), content management systems (CMS), and open source software (Robertson and Robertson, 2012). The IT infrastructure may comprise of business software such as components off the shelf (COTS), open source software and legacy systems that typically cannot communicate with each other in order to share data. The business software often contains identical data stored in multiple locations leading to inefficiencies. The use of COTS based applications (CBAs) has risen from 28% in 1996-97 to 70% in 2001-2002 (Yang, Bhuta, Boehm, & Port, 2005; Selby, 2007 p. 556; Nanz, 2010 p. 8). COTS software products are challenging to integrate into a business process. Boehm (2006a, 2006b) identifies that future software engineering processes will evolve to support the integration of the business software in the IT infrastructure.

The Software Requirements knowledge area is a discipline of software engineering that is concerned with requirement elicitation, analysis, specification and validation of software requirements (Bourque & Fairley, 2014). As part of the Software Requirements knowledge area, today's requirements engineer has to decide the best strategy for identifying the software that fulfils the business needs and aligning that software with the organisation's IT infrastructure. Strategies may include process models that are designed to support the requirements engineer in providing guidance on the order of the activities and creation of artefacts that a project performs (Boehm, 1988). Requirements Elicitation is one of the primary activities in a process model that involves discovering requirements to solve the problem. The Requirements Elicitation activity involves intense communication between stakeholders (Coughlan and Macredie, 2002). Stakeholders are anyone interested in, or affected by the outcome of the future system. Stakeholders typically include business executives such as sponsors of the project, IT architects that build the system, business users that will operate the future system and requirements

engineers that gather the requirements of the future system. Requirements are defined as a condition or capability defined by stakeholder to solve a problem or achieve an objective (International Institute of Business Analysis, 2009). It may aim to automate part of a task in a business process (Bourque & Fairley, 2014).

The stakeholders often find it difficult to articulate their requirements (Nuseibeh & Easterbrook, 2000). Poor communication makes discovery of the requirements a challenge. The use of elicitation techniques such as interviews, group work, prototypes, models and scenarios may provide a solution to this challenge (Bourque & Fairley, 2014; International Institute of Business Analysis, 2009; O'Loughlin, 2010). Sutcliffe (1997) suggests the combination of requirement elicitation techniques such as the use of prototype artefacts and scenarios. Scenarios describe the functionality of a business process through concrete examples of real life sequence of interactions among the main component based applications. Another useful elicitation technique is modelling. Modelling deals with understanding an organisation through the creation of enterprise models of the business process and IT infrastructure.

Requirements Elicitation is a highly collaborative process (Azadegan et al, 2013). Companies use a mix of collaboration technologies such as phone, video/web conferencing, email and group support systems (GSS) to collaborate (Carrmel (1999) in Damian & Zowghi, 2003). Synchronous technologies such as phone and video conferencing have limitations for when stakeholders can take calls because of time differences across the world. According to de Vreede, Briggs, and Massey (2009) such technologies have not seen wide-spread implementation as they require business users to have extensive knowledge about how to use technology to invoke, sustain, and change useful patterns of collaboration. De Vreede and Briggs (2005) developed a collaboration engineering design approach to creating collaborative processes. As an approach it uses general patterns of collaboration to classify group activities based on how a group moves towards a goal (Briggs, Kolfschoten, Gert-Jan, & Douglas, 2006). With the influences of global software development these collaborative processes should be designed for asynchronous use.

Requirements Elicitation is characterised by frequent communication (Coughlan & Macredie, 2002). Frequent communication depends on the richness of the communication channel such as face to face or email. Effective communication has been difficult to achieve and is a recurring problem in the elicitation of requirements (Saiedian

and Dale, 2000; Damian and Zowghi, 2002; Coughlan and Macredie, 2002; Walia and Carver, 2009). Coughlan and Macredie (2002) present a communication framework that promotes effective communication for an organization and its stakeholders in attempting to integrate technology. The challenge is to create a richer channel for communication that embeds effective communication techniques for mediating the communication of requirements.

A prototype is useful for eliciting non-functional requirements (Ameller, Ayala, Cabot, & Franch, 2013). A prototype is the process of developing a scaled down version of the IT infrastructure (Asur and Hufnagel, 1993). It is a quick implementation that supports understanding of functionality and characteristics of the IT infrastructure. Non-functional requirements capture conditions under which the future system should operate or qualities that the future system must have (International Institute of Business Analysis, 2009) in contrast to functional requirements that describe the functions a software component executes. Interoperability is an example of a non-functional requirement in which the future system should be able to work with other systems in the IT infrastructure. Considering the timetabling system, an example of an interoperable requirement is as follows: "the timetable system should interface with the finance system in order to automate the payment to lecturers for each timetabled class". Non-functional requirements usually constrain the system as a whole rather than individual services offered by the system (Sommerville, 2010). As such, non-functional requirements guide the shape of the overall IT infrastructure (Kazman & Bass, 1994; Kruchten, Capilla, & Duñeas, 2009). In essence a functional requirement describes what the future system should do and a non-functional requirement describes how the future system works. Nuseibeh and Easterbrook (2000) identify an open question about how to analyse what impact a particular IT infrastructure choice has on the evolution of requirements. The challenge is to generate a prototype that allows the analysis of an architectural choice that satisfies constraints imposed by non-functional requirements.

Modelling approaches such as enterprise modelling are used as drivers to prompt further requirements elicitation (Nuseibeh and Easterbrook, 2000). Enterprise modelling is often used to capture the purpose of a system, by describing the behaviour of the organisation through the business processes and services that it provides (Greenspan & Feblowitz, 1993). Chandrasekaran, Silver, Miller, Cardoso, & Sheth, (2002) recognised that there are synergies between models and web services. Plugging real web services that form

part of a business process into a model provides an authentic representation with the IT infrastructure. The challenge is to create enterprise models that form the basis of aligning the business process with the IT infrastructure.

In this thesis, it is argued that a collaborative requirements elicitation framework to solve the alignment of the business process and IT infrastructure should comprise a combination of collaboration, communication, prototyping and modelling activities.

1.2   Research Question

The research question posed in this thesis explores to what extent support for shared understanding amongst stakeholders can be fostered through a collaborative requirements elicitation framework. This thesis proposes a combination of collaboration, communication, prototyping and modelling techniques to realise this framework.

In this thesis a **framework** [2] [3] is defined as a description of a process model [4] and tool support. Adapted from Lind and Zmud [5] (1991), this thesis defines **fostering shared understanding** as the degree of mutual understanding between the requirements engineer and stakeholders in order to achieve alignment of the business process and IT infrastructure. Adopted from Robertson and Robertson (2013), this thesis defines **stakeholders** as anyone interested in, or has an effect on the outcome of the future system which shall include a core team containing business executives, IT architects, business users and a requirements engineer. Adopted from Van Lamsweerde (2009), this thesis defines **requirements elicitation** as the discovery of requirements that will shape the future system, based on the weakness of the current system as they emerge from domain understanding. Adopted from Vreede and Briggs, (2005), this thesis defines

---

[2] Definition of a Framework "a set of ideas or facts that provide support for something" [Source http://www.merriam-webster.com/dictionary/framework]

[3] Definition of Framework "In general, a framework is a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful." [Source http://whatis.techtarget.com/definition/framework]

[4] Sommerville describes a process model as a framework which is an abstraction of a process that can be used to explain different approaches to requirements elicitation where the framework that may be extended and adapted to create more specific processes (Sommerville, 2010 pg. 29).

[5] Lind and Zmud (1991) research was to determine if a convergence in understanding between providers and users of a technology would result in greater innovativeness regarding that technology. Two mechanisms were proposed for achieving greater convergence: (1) more frequent communication and (2) the use of richer communication channels. Convergence represents the degree of mutual understanding between the technology providers and the other business personnel about the firm's business activities and the importance of the technology in supporting those activities.

collaboration as joint effort towards a group goal. Adapted from Johnson and Lederer (2006), Coughlan and Macredie (2002), and Rogers and Kincaid (1981), this thesis defines **effective communication** [6] as a dynamic process of idea and knowledge generation, which occurs over time through collaborations with stakeholders such as business executives, IT architects, and business users as they elicit and validate requirements using elicitation techniques such as interviews, workshops, prototypes, scenarios and models which leads to shared understanding and collaborative action. Adapted from Asur and Hufnagel, (1993), this thesis defines **prototyping** [7] as the process of developing a scaled down version of the IT infrastructure. It is a quick implementation that supports understanding of functionality and characteristics of the IT infrastructure. Adapted from Loucopoulos & Kavakli, (1995) and Greenspan & Feblowitz, (1993), this thesis defines **modelling** [8] of enterprise knowledge which incorporates different viewpoints that provide insights to capture the purpose of a future system, by describing the behaviour of the organisation through the business processes and services that it provides.

1.3 Research Objectives

To address the research question, the following specific sets of research objectives were derived:

---

[6] Coughlan and Macredie (2002) concluded with four recommendations for effective communication for an organization and its stakeholders in attempting to integrate technology namely, include business users in the design; select an adequate mix of IT and business users who then interact on a cooperative basis; the incorporation of communication activities that relate to knowledge acquisition, knowledge negotiation and user acceptance; the use of elicitation techniques for mediating communication for the requirements of a system. Rogers and Kincaid (1981) describes communication as a dynamic process of idea and knowledge generation, which occurs over time through interaction with others and which leads to mutual understanding and collective action. Johnson and Lederer (2006) showed that frequent communication between the Chief Executive and Chief Information Officers helped create a shared understanding of the current and future role of IT in the organisation and they also reported more frequent communication using communication channels such as face to face and email that were perceived to be richer.

[7] Definition of a prototype as appeared in Asur and Hufnagel, (1993) is the process of developing scaled down system's versions, before building the delivered system. Software prototyping is an incomplete initial version or pilot model of the intended system. It may be missing functionality, or it may not meet non-functional requirements, such as performance, safety, and cost. A prototype is generally a low cost system representation, used to clarify and to complete requirements by displaying selected aspects of the proposed system.

[8] Loucopoulos & Kavakli (1995) advocate the explicit modelling of enterprise knowledge according to a structured framework which incorporates different viewpoints that provide insights into the purpose of the system, its operational characteristics and its implication on the roles of the different actors affected by the system itself. Of modelling, Nuseibeh & Easterbrook (2000) say "Enterprise modelling is often used to capture the purpose of a system, by describing the behaviour of the organisation in which that system will operate (Loucopoulos & Kavakli, 1995). This behaviour can be expressed in terms of organisational objectives or goals and associated tasks and resources (Yu, 1997). Others prefer to model an enterprise in terms of its business rules, workflows and the services that it will provide (Greenspan & Feblowitz, 1993)"

1 Investigate the state of the art broadly around requirements engineering and specifically requirements elicitation theory that includes background on the techniques that are combined in this thesis to foster shared understanding. The investigation includes formal knowledge representation techniques and controlled natural language to communicate requirements and model a business process scenario. Collaboration and communication as part of shared understanding is also investigated.

2 Design a collaborative requirements elicitation framework that fosters shared understanding between stakeholders as they align business process and IT infrastructure. This framework is called Collaborative Requirements Elicitation through Shared Understanding and Scenarios (CRESUS).

3 Implement the framework and produce a new supporting tool. This tool is called CRESUS-T.

4 Evaluate the collaborative requirements elicitation framework using a case study. At the National College of Ireland (NCI), the IT department undergo a manual business process for rolling out the IT software image to desktop computers on an annual basis. The IT software image contains the software that is required for each module on all programmes offered. An IT administrator is responsible for initiating the business process when they send an email to all lecturers in each school around March or April. Lecturers in each school have four to six weeks where they can review the software that they require for the next academic year and return a software requirement list to the IT administrator. However, the software requirement list never comes back on the due date. The IT administrator sends a reminder email at least three times and sometimes it is escalated to the Dean of School. In this thesis, the NCI manual business process for creating the IT software image forms a case study for the investigation of a collaborative requirements elicitation framework that supports the alignment of the business process and IT infrastructure.

1.4 Contributions

The major contribution of this research is a collaborative requirements elicitation framework called CRESUS. This framework consisting of a process model and tool

support, introduces a novel way of combining techniques to foster shared understanding that differs significantly from current requirement elicitation frameworks. The framework specifically combines collaboration, communication, prototyping and modelling.

Process models in Requirements Engineering provide general guidance on the order of the activities and creation of artefacts that are human-centred and error-prone. Machine supportable processes are largely unexploited in Requirements Engineering. Through the lens of formal knowledge representation, the CRESUS framework provides a formalised process model that is machine understandable. The process model follows the collaborative engineering design approach to creating collaborative patterns that support the stakeholders as they elicit requirements. The stakeholders comprise of business executives, IT architects and business users and are purposefully selected to support effective communication with the framework. Reference architectures that simulate the IT infrastructure are utilised to support the business process. The CRESUS framework is specifically designed to support the alignment of the business process and IT infrastructure.

Another contribution of the research is the new support tool called CRESUS-T as a proof of concept that demonstrates and validates the CRESUS framework.

1.5 Thesis Overview

This thesis proposes an innovative collaborative requirements elicitation framework CRESUS to foster shared understanding amongst stakeholders involved in the alignment of business processes and IT infrastructure.

A review of the state of the art broadly in requirements engineering and specifically requirements elicitation theory is presented in chapter 2. This includes background on elicitation techniques that are combined in this thesis to foster shared understanding. Knowledge about the application domain is examined with particular emphasis on formal knowledge representation techniques. This leads into an examination of controlled natural languages with emphasis on the tools that utilise semantic inference to allow stakeholders communicate requirements in a consistent and understandable manner. A review of the research into business process modelling is conducted to allow stakeholders' model a scenario of collaboration between humans and systems. A state of

the art review is conducted around collaboration and communication techniques that are combined in this thesis to foster shared understanding.

The design of the CRESUS framework that comprises of a process model with four activities is presented in chapter 0. The creation of the knowledge representation artefacts such as the requirements model, IT infrastructure model and business process model are also discussed. A discussion on the design of the support tool, CRESUS-T that implements the two central activities of the process model is also presented. The chapter discusses the main architectural components of CRESUS-T.

Chapter 0 describes the implementation of the CRESUS process model and the technical implementation of the support tool, CRESUS-T. The support tool is responsible for allowing stakeholders communicate and validate requirements, then automatically generate a prototype of the IT infrastructure and finally model a scenario that supports the alignment of the business process and IT infrastructure. The technologies that CRESUS-T uses and the integration of third party services are also discussed in this chapter.

Chapter 5 presents the evaluation of the CRESUS framework. It includes details of the three experimental studies that were undertaken through a stakeholder-based case study within a third level institution. The first experimental study motivates the desire for a framework that supports communication and modelling among senior and middle managers. The second and third experimental study examines the extent the CRESUS framework supports collaborative requirements elicitation.

This thesis concludes with a description of the objectives and achievements of this research. A summary of key contributions to the state of the art of requirements engineering which are attributed to the proposed CRESUS framework in this thesis are provided. Finally a discussion of future research directions is provided.

2   State of the Art

2.1   Introduction

Traditional requirements engineering tends to involve the requirements engineer meeting with the end user to identify the requirements, write the requirement specification, and hand them to the development team to develop the software in-house (Robertson and Robertson, 2012). Today, the majority of software is not developed in house but is available for purchase or free, such as customer resource management (CRM), content management systems (CMS), open source software (Robertson and Robertson, 2012). Today's requirements engineer has to decide the best strategy for identifying the software that fulfils the business needs and aligning that software with the organisations IT infrastructure. The elicitation of requirements to foster shared understanding among stakeholders is a recurring problem in the alignment of the IT infrastructure with the business needs. Boehm (2006a, 2006b) identifies that future software engineering processes will evolve to support the alignment of the IT infrastructure.

A formalised approach to requirements engineering is emerging with the advent of semantic web technologies. Activities within a requirement process such as requirement elicitation provide general guidance on combining elicitation techniques to gather requirements but the treatment of a formalised and systematic way of combining such techniques is limited. Collaboration and communication are important features that may lead stakeholders to a shared understanding of the requirements. However collaboration and communication becomes more challenging with the move away from developing software in house to global software development where asynchronous tools become prominent.

This chapter first in section 2.2 investigates requirements engineering theory in particular focusing on the role formal knowledge representations have played in one of the key activities that is of interest, Domain Understanding. This includes an analysis of requirement elicitation techniques with a focus on combining such techniques. Also included is an analysis of models and how they could support the alignment of the business process and IT infrastructure. Second in section 2.3, shared understanding is examined from a business IT alignment perspective paying particular attention to the role collaboration and communication play in supporting stakeholders involved in global software development.

2.2   Requirements Engineering

Current requirement engineering literature provides general guidance on requirements elicitation as part of the software engineering process such as IEEE Computer Society guide to the Software Engineering Body of Knowledge (SWEBOK), Version 3.0 (Bourque & Fairley, 2014), Software Engineering: A Practitioner's Approach (Pressman & Maxim, 2014), Software Engineering: A Methodical Approach (Foster, 2014), Software Engineering (Mishra & Mohanty, 2011), Software Engineering (Sommerville, 2010), International Institute of Business Analysis guide to the Business Analysis Body of Knowledge (BABOK) Version 2.0 (International Institute of Business Analysis, 2009), and Software Engineering: Principles and Practice (Vliet, 2008).

Requirement engineering literature that focuses exclusively on requirements engineering, providing general guidance on process, methods, tools and techniques based on research and industry practice are:- Volere Method (Robertson and Robertson, 2012), Certified Professional Requirement Exam (Pohl & Rupp, 2011; "International Requirements Engineering Board," n.d.), System goals to UML models (Van Lamsweerde, 2009), and general requirements engineering (Wiegers & Beatty, 2013; Hull, Jackson, & Dick, 2010). The requirements engineer is free to adapt the process, methods, tools and techniques that they are familiar with (Robertson & Robertson, 2012 p. 7). Through collaboration and communication the stakeholders gather the requirements. However, the majority of errors identified in future system, are attributed to requirements that may be incomplete, ambiguous and misinterpreted.

The essence of requirements engineering is the need to understand and define what the right problem is, why it needs to be solved, what the future system should do, what stakeholders should be involved, and how the use of the future system can support the business needs (Van Lamsweerde, 2009).

Requirements engineering is a human-centred process (Nuseibeh and Easterbrook, 2000). As such, understanding the requirements of a problem is a challenging activity for a requirements engineer (Pressman, 2010). A requirement **process model** is one such support that is available to help with that challenge.

The requirements process varies depending on the type of software application being developed. It can range from a requirement process used in a major organisation to

guidelines on techniques such as a brainstorming session (Sommerville, 2005). A process model is a simplified representation of a process at an abstract level that provides a framework for scaffolding the actual process (Sommerville, 2010).

Boehm, (1988) indicates that "Process models provide guidance on the order of the activities and creation of artefacts that a project performs". IEEE Computer Society (Bourque & Fairley, 2014) describes how the activities of elicitation, analysis, specification and validation are configured for different projects. The BABOK guide[9] (International Institute of Business Analysis, 2009) describe knowledge areas that may represent activities in a process such as business analysis planning and monitoring, elicitation, requirements management and communication, enterprise analysis, requirements analysis, and, solution assessment and validation. The BABOK is a guide and should not be construed as a methodology. Van Lamsweerde (2009) describes similar activities that compose a requirements process such as domain understanding, requirements elicitation, evaluation and agreement, specification and documentation, and requirements consolidation. In essence, the activities of a requirements process are similar. Following a requirement process is challenging and is limited by the ability of the requirements engineer to think (Robertson & Robertson, 2012). The research tends to focus on general activities that may be adapted rather than a formalised approach to process activities that could be supported by machine understanding.

This section will analyse the individual activities of the process model through the lens of the emerging trend based on a formal knowledge representation perspective as discussed. The lens will specifically focus on Domain Understanding (see section 2.2.1), Requirements Elicitation (see section 2.2.2) and, Modelling and Analysis (see section 2.2.3).

2.2.1 Domain Understanding

According to Van Lamsweerde (2009 p. 61), the goal of domain understanding is to understand the problem, and the application domain of the problem. The IEEE Computer Society (Bourque & Fairley, 2014) indicates that the requirements engineer needs to acquire and structure available knowledge about the application domain of the problem. Recording this application domain knowledge makes communication easier and future understanding more reliable (Robertson and Robertson, 2012). The degree and manner

---

9 IIBA® announced upcoming launch of BABOK® Guide v3 for April 15, 2015

of specifying application domain knowledge varies and is dependent on the formality used. Glossaries and data dictionaries are informal and reside at one end of the spectrum. Moving to the other end are formal knowledge representations such as ontologies. Ontology is understood as a way of structuring and specifying the meaning of knowledge in an application domain of the problem (Allemang, 2008 in Castañeda, Ballejos, Caliusco, & Galli, 2010). The IEEE Computer Society (Bourque & Fairley, 2014) indicates that it is good practice to follow an ontological approach.

Gruber (1993) defines an ontology "as an explicit specification of a conceptualisation". In the context of artificial intelligence, an ontology is referred to as an engineering artefact that is based on the vocabulary consisting of terms and relationships of an application domain in addition to the rules for combining the terms and relationships between the terms (Guarino, 1998; Neches et al., 1991). The motivation for using ontology is that it allows stakeholders and other systems have a shared understanding of the structure of information (Gruber, 1993; Musen, 1992). In addition, they are machine-interpretable and are amenable to semantic analysis.

Some of the early work on ontologies in requirements engineering centred around knowledge representation languages such as Requirements Modelling Languages (Greenspan, 1984; Greenspan, Mylopoulos, & Borgida, 1994), Telos (Mylopoulos, Borgida, Jarke, & Koubarakis, 1990), KAOS (Dardenne, van Lamsweerde, & Fickas, 1993), and the i* framework (Yu & Mylopoulos, 1994). The emergent research area of the semantic web has facilitated a renewed interest in the adoption of ontologies for requirements engineering. The semantic web builds structure and meaning to data that is web accessible (Berners-Lee, Hendler, & Lassila, 2001). Semantic Web technologies such as XML (Bray, Paoli, Sperberg-McQueen, Maler, & Yergeau, 2008), RDF ("RDF - Semantic Web Standards," 2014) and OWL ("OWL 2 Web Ontology Language Document Overview (Second Edition)," 2012) enable stakeholders to create requirements on the Web, build vocabularies, and write rules for handling the requirements. XML adds arbitrary structure to the stakeholder's documents but does not describe what they mean. Meaning is expressed by RDF, using sets of triples, where each triple is like the subject, verb and object of an elementary sentence.

The benefit of ontologies for requirements engineering, is the ability to explicitly model domain knowledge in a machine interpretable way. Ontologies provide reduced ambiguity, increased meaning, formality and support for automated reasoning. This

allows for requirements traceability, consistency checking, in addition to automatically generating the software specifications (Dobson & Sawyer, 2006).

Dermeval et al., (2015) conducted a systematic literature review to identify how ontologies support the activities of the requirements process such as elicitation, analysis, specification, validation and management of requirements. Two studies (Al Balushi, Sampaio, & Loucopoulos, 2013; Anwer & Ikram, 2008) out of sixty seven addressed all the activities of the requirements process. Al Balushi, Sampaio, & Loucopoulos, (2013) discussed a quality-driven Requirements Engineering (RE) framework and tool that applies knowledge management techniques and quality ontologies to support RE activities. Anwer & Ikram, (2008) proposed a unified GORE process that defines roles, activities, and artefacts and their relationships. However, building a formal knowledge representation of the application domain was outside the scope of their work. Dermeval et al., (2015) suggested that the use of ontologies to support the entire requirements elicitation process needs to be further investigated.

Castañeda, Ballejos, Caliusco, & Galli, (2010) comprehensively reviewed and presented the use of ontologies in requirements engineering. Formalised knowledge representation approaches tended to concentrate on the structure of requirements specification documents (Castañeda, Ballejos, & Caliusco, 2012; Groza, Schutz, & Handschuh, 2007), representing requirements (Siegemund, Thomas, Zhao, Pan, & Assmann, 2011; Lin, Fox, & Bilgic, 1996), managing requirements (Veres, Sampson, Bleistein, Cox, & Verner, 2009), dependability between requirements (Dobson & Sawyer, 2006) and representing application domain knowledge (Li, Jin, Xu, & Lu, 2011; Fonseca, 2007; Guarino, 1998).

Guarino (1998) distinguished between *ontology-aware* and *ontology-driven systems* (Yildiz & Miksch, 2007). An ontology-aware system knows of the existence of an ontology and can query the ontology. In an ontology-driven system, which is the focus of this research, the ontology is placed as a component within the system. The reason for using an ontology-driven system is that it enables communication through messages that contain expressions formulated from the ontology. Guarino (1998) refers to this as *ontology-driven communication*. The advantage of this approach is that machines can automatically reason over and understand the communication. The ontology can impact the components of a system such as the application program, user interface and database.

In addition, Guarino (1998) first describes the use of the ontology to develop the static part of the program in the form of type or class declarations and procedures. The advantage of this approach is that it ensures that domain knowledge is represented in the application program. Abbott's (1983) technique for developing software programs from informal English descriptions to derive data types from common nouns, objects from proper nouns and operators (functions, procedures) from verbs, provides insight in how to generate software programmes from informal but precise English. Abbott's technique demonstrates that in a highly automated process, a formal knowledge representation model which is complete and precise is amenable to automatic generation of the software programmes (Berzins, Martell, Luqi, & Adams, 2008). Booch extended Abbott's technique to object oriented development (Booch, 1986). Secondly, in the context of the user interface components, Guarino describes an ontology as the embodiment of semantic information on the constraints imposed on the classes and relationships from the application domain. Using the ontology for the user interface allows for semantic checking of any violations on those constraints. Thirdly, Guarino (1998) discusses using ontologies for databases where ontologies that integrate lexical resources like WordNet may support the analysis of natural language requirements. Such ontologies can be mapped to schemas for different types of databases (Ceri & Fraternali, 1997 in Guarino, 1998).

In addition to domain knowledge, Loucopoulos and Champion (1988) argue for method knowledge from the model representations perspective, as an approach to requirements engineering in order to provide tools to support the requirements engineer. Models may be represented by their diagrams, for example in entity life history analysis a developer may use an entity life history diagram. There is a clear separation of concerns between the model, and the representation of the model through a diagram. So for example an entity life history diagram may be represented in natural language instead of a diagram without any loss of information. The advantage with such a linguistic approach is that the application domain knowledge is presented to the stakeholder through the semantics of the domain rather than the semantics of a design model.

The difficulty in understanding knowledge-based techniques is a well-known issue with ontologies. This is only made worse in requirements engineering since requirements are aimed at a range of stakeholders with different backgrounds and knowledge (Dobson & Sawyer, 2006). Natural language is one of the best mediums for communicating and

understanding requirements. This allows stakeholders and especially business executives who may be unfamiliar with modelling notations to still understand and validate requirements. However, natural language is ambiguous and can be easily misinterpreted (Church & Patil, 1982).

If the language is *controlled*, that is, the system prompts the user on the construction of the sentences, then the language becomes machine readable. By incorporating semantics, then other systems will have the same meaning for the sentences that are created. A *controlled natural language* (CNL) is a precisely defined subset of full English that can be used for communicating the organisation's needs in such a way that it can be automatically reasoned over by machines and thus removes the ambiguity issues of natural language.

Pool, (2006) categorises controlled natural languages under two areas, namely naturalistic and formalistic. Where naturalistic improve readability for human readers, and formalistic enables reliable automatic semantic analysis of the language. Naturalistic CNLs are typically used to increase the quality of technical documentation and formalistic CNLs are used for knowledge representation and reasoning. Pool, (2006) also categorises CNLs as restrictive or general. Restrictive CNLs are designed for limited, intra-organization purposes. General CNLs are designed for multi-domain meaning and expression such as in the Semantic Web.

The approach taken in this thesis is along the lines of formalistic and general controlled natural languages for knowledge representation of ontologies for the semantic web. Pool (2006) surveys forty one controlled languages of which five come under the category of formalistic and general namely Attempto Controlled English (ACE) (Fuchs, Kaljurand, & Kuhn, 2008; Fuchs, Höfler, Kaljurand, Rinaldi, & Schneider, 2005), CPL (Clark et al., 2005), E2V (Pratt-Hartmann, 2003), and Formalised English (Martin, 2002). Of the evaluated languages, ACE appears to have the most active development. Schwitter, Kaljurand, Cregan, Dolbear, & Hart (2008) compared three controlled natural languages that have been designed to express the logical content of OWL ontologies namely, ACE, Ordnance Survey Rabbit (Hart et al., 2008; Dolbear et al., 2007) and Sydney OWL Syntax (Cregan, Schwitter, & Meyer, 2007). They indicate that the goal of these three languages was to make OWL ontologies accessible to stakeholders with no training in formal logics. A review of the state of the art also indicates that PENG-D (Schwitter &

Tilbrook, 2004) and CLoNE (Funk et al., 2007) are additional examples of controlled natural languages.

Of all the controlled natural languages, ACE appears to have been the most studied and provides the most support with additional tools such as APE, RACE, AceRules and an OWL Verbaliser. ACE is a knowledge representation language used in applications that include ontology authoring (Fuchs et al., 2006a; Kuhn, 2006), semantic querying (Bernstein et al., 2004), reasoning (Kuhn, 2007; Fuchs & Schwertel, 2003) and ontology verbalization (Kaljurand & Fuchs, 2007). In addition it has the ability to express business and policy rules, and query tasks on the semantic web. A predictive editor (Kuhn, 2008) guides stakeholders, word-by-word in the construction of a sentence that complies with ACE. The sentence can be converted to an ontological format using the ACE Parsing Engine (APE) (Fuchs, Höfler, Kaljurand, Rinaldi, & Schneider, 2005). APE ensures that the sentence is ACE compliant and then converts the natural language sentence into an OWL representation. The approach described with ACE shows the potential for a natural language interface to utilize semantic inference to refine and hone the stakeholder's requirements.

This thesis proposes to build on the approach to create an ontology-driven process model that is influenced by Yildiz & Miksch (2007) and Guarino (1998) with influences from Fuchs, Höfler, Kaljurand, Rinaldi, & Schneider (2005) using a controlled natural language editor to create requirements based on a formal knowledge representation of the application domain and modelling language. Such an approach should facilitate the creation of the ontologies by stakeholders using natural language. The formal knowledge representation of the application domain facilitates constraining the vocabulary of the natural language and allows ontology-driven communication. A formal knowledge representation is also amenable to automatic machine translation to develop the static part of the software components and database that represent the IT infrastructure. In addition **this thesis proposes to include an ontology around method knowledge with a specific focus on models as described in Loucopoulos and Champion (1988).** This allows stakeholders and especially business executives who may be unfamiliar with modelling notations to still understand and validate the requirements based on natural language. This ensures that the stakeholders have the same understanding of the requirements and that the requirements are traceable,

consistent and not ambiguous. In addition, such an approach is **amenable to automatic machine translation to generate a diagrammatic representation of the model**.

2.2.2 Requirements Elicitation

Requirements Elicitation is an activity that involves discovering requirements to solve the problem in hand (Bourque & Fairley, 2014; International Institute of Business Analysis, 2009). This activity is characterised by intense communication among stakeholders (Coughlan and Macredie, 2002). It is generally recognised that stakeholders find it difficult to articulate their requirements (Nuseibeh & Easterbrook, 2000). In addition, the tasks that stakeholders perform may be so ingrained in their work habits, that they forget how they are done (Robertson and Robertson, 2012). How to discover the requirements from these difficult situations is a challenge.

The use of requirements elicitation techniques provides a solution to this challenge (Bourque & Fairley, 2014; International Institute of Business Analysis, 2009). Elicitation techniques include interviews (Agarwal & Tanniru, 1990), group work (Wood & Silver, 1995; Takeda, Shiomi, Kawai, & Ohiwa, 1993), prototyping (Budde, Kautz, Kuhlenkamp, & Züllighoven, 1992), scenarios (Alexander and Maiden, 2004; Hsia, Samuel, Gao, Kung, Toyoshima and Chen, 1994; Holbrook, 1990) and modelling (Bourque & Fairley, 2014).

Several studies have surveyed elicitation techniques in order to classify them (Dieste & Juristo, 2011; Davis, Dieste, Hickey, Juristo, & Moreno, 2006; Zowghi & Coulin, 2005; Nuseibeh & Easterbrook, 2000; Maiden & Rugg, 1996; Hoffman, Shadbolt, Burton, & Klein, 1995; Cooke, 1994; Boose, 1989). Nuseibeh and Easterbrook (2000) provide one such classification of requirements elicitation techniques based on traditional techniques, group elicitation techniques, prototyping techniques, model-driven techniques, cognitive techniques and contextual techniques as described in Table 2-1.

Table 2-1 Classification of Requirement Elicitation Techniques

Classification	Description	Technique
Traditional	Generic data gathering techniques	Questionnaires
		Surveys
		Interviews

		Analysis of existing documentation such as organisational charts, process models, standards, manuals of existing systems
Group	To foster stakeholder agreement, buy-in and exploit team dynamics to elicit a richer understanding of needs	Brainstorming
		Focus groups
		RAD/JAD workshops
Prototype	Used where great deal of uncertainty exists about the requirements. Readily combined with other techniques.	Throwaway
Model-driven	Specific model of the type of information to be gathered.	Goal-based methods
		Scenario-based methods
Cognitive	Developed for knowledge acquisition for knowledge based systems	Protocol Analysis
		Laddering
		Card sorting
		Repertory grids
Contextual	Bringing the analyst into the stakeholders world	Ethnographic - participant observation
		Ethnomethodology
		Conversation analysis

Consensus exists that one elicitation technique cannot work for all situations. A combination of techniques will often allow the requirements engineer get a picture of the future system. Almost all requirements books describe multiple elicitation techniques (Robertson and Robertson, 2103; Pressman, 2010, Sommerville, 2010; Van Lamsweerde, 2009). However they only offer a toolbox of techniques and guidance for the requirements engineering process. As an example Robertson and Robertson (2013,) indicate that the interview should be used in conjunction with the use of other elicitation techniques. But do not elaborate any further.

Zowghi & Coulin (2005) indicate that it is useful to select elicitation techniques that are complementary to each other. Table 2-2 provides guidance on the elicitation techniques that can be used in cooperation (marked with a "C") or as an alternative (marked with an "A") to each other. The table is not exhaustive and only presents a selected group of techniques that are relevant to this thesis. Their work is largely based on their assessment of the literature as well as practical experience and observation in requirements

elicitation research and practice. However, it is not clear what empirical evidence backs up this assessment.

Table 2-2 Complimentary and Alternative Elicitation Techniques

	Interviews	Group-work	Prototyping	Goals	Scenarios
Interviews		A	A	C	C
Group-work	A		C	C	C
Prototyping	A	C		C	C
Goals	C	C	C		C
Scenarios	C	C	C	C	

The literature also presents a combination of elicitation techniques as prescribed by a specific methodology that is being followed such as Rapid Application Development (Martin, 1991), collaborative requirements elicitation and validation model (Dean, Lee, Pendergast, Hickey, & Nunamaker Jr., 1997).

Dean, Lee, Pendergast, Hickey, & Nunamaker Jr. (1997) present a collaborative requirements elicitation and validation model (CREV). CREV combines activity, data, and scenario models with prototypes to determine requirements. They also define a model of user communication, showing the levels of user participation for different elicitation techniques. **Their work is similar to this thesis. However this thesis distinguishes itself through the use of formal knowledge representation models.**

Dieste & Juristo, (2011) conducted a systematic review on elicitation techniques. They examined 564 publications of which they selected 26 publications. The 26 publications contained 30 empirical studies. This work appears to be based on an earlier publication by Davis, Dieste, Hickey, Juristo, & Moreno (2006) and is limited to March 2005. An analysis of their findings indicates that two publications relate to combining elicitation techniques. One publication by Frankel (1987) is in the area of knowledge elicitation and is deemed not relevant to this research. The remaining publication by Sutcliffe (1997) is relevant.

Sutcliffe (1997) proposes Scenario-based Requirements Analysis Method (SCRAM). SCRAM combines several techniques that involve early prototyping, scenario-based

analysis and design rationale. The approach he takes is to demonstrate a prototype that is situated in a scenario context of use, combined with systematic questioning and exposure of the designer reasoning process. The techniques are combined with a method to provide process guidance such as advice on setting up sessions, use of the techniques, and detailed guidance on gathering and validating the requirements. The method encourages collaboration between two to three business users and two requirements engineers. A scenario is developed based on the preliminary domain analysis consisting of half to one page narrative that runs through the key tasks of the user. The prototype was a menu driven user interface that a user interacted with through a series of point and clicks. Their approach generated a large number of requirements and facts and user participation was successful. Their results indicate that the approach of combining techniques was very effective in eliciting requirements.

Sutcliffe (1997) suggests several guidelines from their experience that relate to this thesis. The first guideline is that the combination of techniques is helpful and use of an artefact such as a prototype is advisable. The second guideline indicates that use of a scenario helps focus business users' attention. One issue that arose in their research was that the difference in how the requirements engineer applied the method and questioning techniques affects the consistency of the results. The third guideline addresses this issue. The guideline suggests training the requirements engineer on the method and questioning technique. Although the third guideline of training the requirements engineer on method and questioning techniques is superior to ensuring consistency of requirements elicitation, there is still no guarantee this will work. **This thesis proposes to incorporate Sutcliffe's (1997) first and second guidelines of combining techniques such as the use of a prototype in the context of a scenario.**

Ameller, Ayala, Cabot, & Franch (2013) in a survey of software architects found that a prototype was useful for eliciting non-functional requirements. They observed that the software architects were the main source of non-functional requirements. It is recognised that non-functional requirements guide the shape of the software architecture (Kruchten, Capilla, & Duñeas, 2009; Kazman & Bass, 1994). Nuseibeh and Easterbrook (2000) outline a roadmap in which there is a need for better understanding of the impact of software architectural choices on the prioritisation and evolution of requirements. While work in software architectures has concentrated on how to express software architectures and reason about their behavioural properties, there is still an open question about how to

analyse what impact a particular architecture choice has on the ability to satisfy current and future requirements. The architectural choices may be guided by reference architectures. Reference architectures represent models of domain specific software structures (Berzins, Martell, Luqi, & Adams, 2008). They provide a template solution for architecture of a particular domain.

The influences from the analysis of non-functional requirements indicate a need to include reference architecture as the basis of the prototype to facilitate the elicitation of non-functional requirements. **This thesis proposes to distinguish its use of prototypes not by focusing on user interfaces with point and click but by creating a prototype of the IT infrastructure that allows the analysis of an architectural choice.**

2.2.3 Modelling and Analysis

Nuseibeh and Easterbrook (2000) indicate that modelling and analysis is an activity that involves the construction of abstract descriptions that are amenable to interpretation. Modelling approaches are used as elicitation tools, where the modelling notation and partial models are used as drivers to prompt further requirements elicitation (Nuseibeh and Easterbrook, 2000).

Stakeholder requirements are, typically, described in natural language and are focused in the application domain. System requirements result from the requirements engineer's effort to represent and organise the stakeholders' viewpoint of requirements at the application domain. These requirements comprise of conceptual models and constitute the first representation of the future system (Machado, Ramos, & Fernandes, 2005). Conceptual models play an important and inexpensive role in displaying significant features and characteristics of dynamic system, which one wishes to study, predict, modify, or control (Kellner, Madachy, and Raffo, 1999).

The choice of modelling methods and analysis techniques depends on the context of the future system. There are several modelling approaches such as enterprise modelling, data modelling, behaviour modelling and domain modelling (Bourque & Fairley, 2014; Nuseibeh & Easterbrook, 2000). The context for enterprise modelling is the organisation in which the future system will operate. It is this context that is of interest in this thesis.

Enterprise modelling is often used to capture the purpose of a system, by describing the behaviour of the organisation in which that future system will operate (Loucopoulos &

Kavakli, 1995). While some authors model organisational goals, tasks and resources (Yu, 1997) others such as Greenspan & Feblowitz (1993) model an enterprise from a viewpoint perspective through the business processes and services that it provides. Viewpoints represent an approach to creating a complete conceptual model from multiple sources of information (Bagheri & Ghorbani, 2008; Andrade et al., 2004; Nuseibeh, Kramer, & Finkelstein, 1994). Nuseibeh, Kramer, & Finkelstein, (1994) defines viewpoints as "loosely coupled, locally managed distributed objects which encapsulate partial knowledge about a system and its domain, specified in a particular, suitable representation scheme, and partial knowledge of the process of development". According to Bagheri & Ghorbani (2008) not all viewpoint-based requirements engineering conform to this definition but mostly agree on the basis of gathering information from multiple sources. Greenspan & Feblowitz (1993) define the elements of a requirement modelling and analysis framework from four viewpoints:-

- As a set of services that meet goals or address the needs of the customer.
- As a set of workflows or business processes performed by an organisation to provide the services.
- As a set of organisational units that service as loci of responsibility for the work.
- As a set of systems that provide the capabilities and resources for performing the work.

Modelling approaches such as Business Process Modelling (BPM) is one of the more popular modelling representation approaches that involve representing business processes of an enterprise, so that they may be analysed (Monsalve, April, & Abran, 2012; OMG, 2011; International Institute of Business Analysis, 2009). The approach taken by Greenspan and Feblowitz (1993) with influences from IEEE Computer Society (Bourque & Fairley, 2014), Object Management Group (OMG, 2011) and International Institute of Business Analysis (2009) is of interest in this thesis.

It is recognised that there are synergies between models and web services (Chandrasekaran, Silver, Miller, Cardoso, & Sheth, 2002). Maiden (2006) argues that requirements engineers will specify requirements for applications that invoke at least some web services. He indicates "It is like having several software prototypes that you can explore for new requirements, demonstrate to stakeholders, and validate requirements with. And they're right there—available to you—through simple queries". As part of an EU funded project, Maiden (2006) ran studies in which requirement

engineers specified requirements for an automotive application using web services that they had discovered. The requirements engineers ranked requirements discovered from web services as more novel than requirements discovered from traditional techniques, such as use-case analysis (Maiden, 2006). One of the approaches that Chandrasekaran, Silver, Miller, Cardoso, & Sheth (2002) propose is the creation of models from web services in order to provide a high fidelity between the model and the IT infrastructure that includes the web services. This approach provides an ability to plug real web services into models, thus creating an alignment between the model and the IT infrastructure that utilises as much realistic data as possible.

This thesis proposes to adopt the viewpoint-based requirements engineering approach with influences from IEEE Computer Society (Bourque & Fairley, 2014), Monsalve, April, & Abran (2012), Object Management Group (OMG, 2011), International Institute of Business Analysis (2009), Maiden, (2006), Chandrasekaran, Silver, Miller, Cardoso, & Sheth, (2002) and Greenspan and Feblowitz (1993), in creating enterprise models from web services that forms the basis of aligning the business process to the IT infrastructure.

The objective of modelling is to communicate understanding of an application domain. Gemino and Wand (2004) present an evaluation framework for conceptual models in which they highlight the use of comprehension tests for capturing the understanding a participant develops when interpreting a diagram. Comprehension tests contain questions about a model that has been removed representing **model recall**. Gemino and Wand (2004) indicate that if the focus of modelling is to understand the application domain then there is also need to focus on **problem solving questions** that require participants to use the mental understanding they have developed from the diagram to suggest answers that were not directly available in the diagram. Gemino and Wand (2004) indicate that "the more answers participants provide, the more sophisticated is their (cognitive) model of the domain and the higher is the level of understanding they have developed". **This thesis proposes to adopt the measures that relate to model recall and problem solving for evaluating the business process to determine if understanding of the application domain has taken place.**

2.3   Shared Understanding

The review of the literature on Shared Understanding was organised into three themes that are grounded in alignment theory, communications theory, and collaborative engineering design theory.

Alignment theory focuses on how IT infrastructure can support the business needs or more specifically the business process. **As requirements elicitation is a highly social process, the approach taken in this thesis will confine itself to the social dimensions of business-IT alignment (Chan and Reich, 2007; Reich and Benbasat, 2000).** The social dimension of alignment refers to the state in which business and IT executives understand and are committed to the business and IT mission, objective and plans. The social dimension is influenced by three factors that are relevant to this thesis. These factors are connection between business and IT planning processes; shared domain knowledge between business and IT executives; and communication between business and IT executives.

Reich and Benbasat (2000) first factor proposes that the level of connection between business and IT planning processes will positively influence the level of alignment. **The approach taken in this thesis is on the connection between the business process and the IT infrastructure.** How the business process is analysed and supported by the alignment of the IT infrastructure was considered in sections 2.2.2 and 2.2.3.

Reich and Benbasat (2000) define the second factor, shared domain knowledge, as the ability of IT and business executives, at a deep level, to understand and be able to participate in the others' key processes and to respect each other's unique contribution and challenges. **In the context of this thesis it is argued that the better that the business executives and IT architects share understanding and participate in each other's key processes, the better the alignment will be.** Shared domain knowledge indicates a very strong understanding of concepts and relationships from the application domain. Shared domain knowledge has been explored in section 2.2.1. A more detailed view of the second factor involves participation in each other's key processes. This implies collaboration which is further explored in section 2.3.1.

Lind and Zmud (1991) provide empirical support for the connection between increased communications leading to increased shared understanding. Reich and Benbasat (2000)

third factor extends the linkage of communication and propose that the level of communication between business and IT executives will influence the level of alignment. This theme of communication is further discussed in section 2.3.2.

2.3.1 Collaboration

Requirements elicitation is a highly collaborative process (Azadegan et al, 2013) that depends on stakeholders fostering a shared understanding of the application domain in order to identify the requirements. De Vreede and Briggs (2005) define collaboration as making a joint effort towards a group goal.

With the influence of global software development, companies use a mix of collaboration technologies such as phone, video/web conferencing, email, social media and group support systems (GSS) to collaborate. Synchronous technologies such as phone and video conferencing usually mean someone at one location will have to change their work-schedules in order to take the calls. However GSS are more suitable to tasks that require concerted reasoning by many minds such as in requirements elicitation (Boehm, Grunbacher, and Briggs, 2001). The primary purpose of a GSS is to support the creation of sustained predictable, repeatable patterns of human interaction and reasoning among people working toward a common goal (Boehm, Grunbacher, and Briggs, 2001).

Damian et al (2000) discussed the challenges of collaborative technologies in multi-site organisations and provided recommendations that impact on collaboration. They provided recommendations to improve the practice of requirements elicitation such as scheduling face to face meetings at the beginning of global projects to establish initial relationships with key stakeholders. Another recommendation was to provide a facilitator and an integrated richer communication media that integrates data, video, and audio channels. This is to address the challenge of ineffective requirements decision making meetings and to manage conflict.

Much has changed since. In the same year as Damian et al's (2000) article, a new research discipline called collaborative engineering was starting to emerge. Collaborative engineering was motivated by the need to create work practices that didn't require trained facilitators. Agres et al. (2005) described the facilitator's skillset as articulate, problem-solving and people oriented employees that are comfortable with technology. People with these skill sets are often promoted or they leave the organization (Agres et

al. 2005). Research in collaboration engineering has focused on designing collaborative work practices for recurring activities that stakeholders can successfully execute for themselves (Briggs, Kolfschoten, Gert-Jan, & Douglas, 2006; de Vreede, Briggs, and Massey, 2009). This approach removes the need for a trained facilitator. De Vreede and Briggs (2005) developed a collaboration engineering design approach to creating collaborative processes. They argue that organisations struggle to make collaboration work and turn to implementing technologies which they say, seldom works. The approach uses general patterns of collaboration to classify group activities based on how a group moves towards a goal (Briggs, Kolfschoten, Gert-Jan, & Douglas, 2006).

Briggs, Kolfschoten, Gert-Jan, & Douglas (2006) identified six patterns of collaboration namely, Generate, Reduce, Clarify, Organize, Evaluate, and Build Consensus. The pattern *Generate* involves moving from having fewer concepts to having more concepts. The *Clarify* pattern involves moving from less to more shared understanding of the concepts and of the words and phrases used to express them. The *Reduce* pattern involves moving from having many concepts to a focus on fewer concepts that the group deems worthy of further attention. The *Organize* pattern involves moving from less to more understanding of the relationship among concepts the group is considering. The *Evaluate* pattern involves moving from less to more understanding of the relative value of the concepts under consideration. The evaluate pattern has three sub patterns namely, Poll, Rank and Assess. Poll assesses the group opinion with respect to the concepts. Rank identifies an order of preference among concepts. Assess specifies and elaborates on the value of concepts. The *Build Consensus* pattern involves moving from having fewer to having more group members who are willing to commit to a proposal (De Vreede, Kolfschoten, & Briggs, 2006).

Patterns of collaboration are created using building blocks called *thinkLets*. A thinkLet is defined as "the smallest unit of intellectual capital to create a pattern of collaboration" (Briggs, De Vreede, & Nunamaker Jr, 2003). A thinkLet is a named, packaged facilitation technique, captured as a pattern that can be incorporated into process designs (De Vreede, Kolfschoten, & Briggs, 2006). Thinklets can be combined to create a sequence of steps for a group to execute in order to achieve collaborative goals. *DirectedBrainstorm*, *StrawPoll* and *MoodRing* are examples of thinkLets (Kolfschoten, Briggs, Appelman, & de Vreede, 2004; de Vreede, Briggs, & Massey, 2009). de Vreede, Briggs, & Massey (2009) describe the purpose of the DirectedBrainstorm thinkLet is to

generate creative ideas in response to prompts from a moderator. This thinkLet is based on the Generate pattern of collaboration. StrawPoll is used to evaluate a number of concepts with respect to one or more criteria. MoodRing is used to track the level of consensus within the group regarding a certain issue.

As an example of usage consider the case where a team want to generate marketing ideas to sell a product. The team meet in a room and come up with ten creative ideas to market the product. They then vote on the top three ideas. There are two patterns of collaboration in this scenario. The first pattern is where team move from no ideas to ten ideas. This represents the Generate pattern. The second pattern is where the group decide on the top three ideas. This represents the Poll pattern. ThinkLets are used to implement a pattern. In this example the two thinkLets, DirectedBrainstorm and StrawPoll can be combined as a sequence of steps, the goal of which is to identify three ideas to market a product. The sequence of steps represents a collaborative process.

Bittner & Leimeister (2013), focus on the systematic development of a reusable process to support groups that lead to a shared understanding. They apply the collaborative engineering design approach to creating a reusable process. The theoretical basis for their design guidelines and design decisions are based on the Van den Bossche et al (2011) model. The model describes team learning behaviours that lead to the construction of shared understanding. The process was validated in a requirements elicitation workshop. One issue that was observed in the workshop was partly related to the support tool not fully supporting the *process activities* which caused an increase in cognitive load on the participants.

For this thesis the influences from collaborative engineering indicate a need to create a process model that influences collaboration among stakeholders as opposed to just using collaborative tools. The activities of the process model should be guided by collaborative patterns such as generate, reduce, clarify, organise, evaluate and build consensus. According to De Vreede, Briggs, and Massey (2009), most collaboration engineering studies have focused on face-to-face settings. They recognise the need to explore the applicability of collaboration engineering concepts to virtual collaboration, as virtual team work and virtual work environments are becoming more dominant. With the influences of global software development these process models should be designed for asynchronous use. In addition, research from Bittner & Leimeister (2013) indicates a

requirement for GSS tool support that facilitates all the activities of the collaborative process.

2.3.2 Communication

Rogers and Kincaid (1981) describes communication as a dynamic process of idea and knowledge generation, which occurs over time through interaction with others and which leads to shared understanding and collective action.

The Requirements Elicitation activity is characterised by frequent communication (Coughlan & Macredie, 2002). However effective communication has been notoriously difficult to achieve and is a recurring problem in the elicitation of requirements (Saiedian and Dale, 2000; Damian and Zowghi, 2002; Coughlan and Macredie, 2002; Walia and Carver, 2009).

The work of Lind and Zmud (1991) showed empirically that frequent communication helped create a shared understanding between technology providers and business users regarding the importance of technology in supporting the business activities. They also demonstrated that communication richness helped create a shared understanding and this was determined by the type of communication channel such as face to face, computer mediated or written documents. Johnson and Lederer (2006) extend the work of Lind and Zmud (1991) to communication between the Chief Executive and Chief Information Officers. Their work showed that frequent communication helped create a shared understanding of the current and future role of IT in the organisation. They also reported more frequent communication using communication channels such as face to face and email that were perceived to be richer.

Coughlan and Macredie (2002) conducted an analysis of effective communication in requirements elicitation. Coughlan and Macredie (2002) concluded with four recommendations for effective communication for an organization and its stakeholders in attempting to integrate technology namely, include business users in the design; select an adequate mix of IT architects and business users who then interact on a collaborative basis; the incorporation of communication activities that relate to knowledge acquisition, knowledge negotiation and user acceptance; the use of elicitation techniques for mediating communication for the requirements of a system such as interviews, brainstorming, prototyping and scenarios.

Inadequate communication and time difference are known problems of global software development (Herbsleb & Moitra, 2001; Damian & Zowghi, 2003). The driver for global software development is outsourcing. Robertson and Robertson (2013) emphasises that outsourcing has become a major part of software requirements. The time difference impacts on the stakeholder's ability to conduct synchronous communication such as face to face. Damian & Zowghi (2003) observed that asynchronous communication becomes the predominant form of communication.

This thesis proposes to extend the work of Rogers and Kincaid (1981), Lind and Zmud (1991), Johnson and Lederer (2006) to requirements elicitation. With the influences of global software development, this thesis proposes to create a rich channel for asynchronous communication to foster shared understanding. The rich channel shall distinguish itself from typical asynchronous communication techniques such as email by embedding techniques for mediating the communication of requirements as discussed in Coughlan & Macredie (2002).

2.4 Summary

This chapter reviewed requirements engineering with particular emphasis on process models that provide guidance on the order of activities and creation of artefacts. Process model activities were examined with requirements identified for structuring a process model around activities that include Domain Understanding, Domain Modelling, Requirements Elicitation, and, Modelling and Analysis in order to provide guidance on gathering, analysing and understanding the requirements. The limitation with current process models is that research tends to focus on general activities that may be adapted. Process models may be enhanced using a formalised approach to process activities that could be supported by machine understanding.

The examination of Domain Understanding identified **ontologies** as a formal method for structuring and representing application domain knowledge and model knowledge. The use of ontologies in requirements elicitation was examined with requirements being identified for applying them to ontology driven systems that support ontology driven communication. Limitations were also identified with ontologies in Requirements Elicitation. These limitations demonstrate that there is little research on the use of ontologies with prototypes.. The use of the prototype elicitation technique for identifying non-functional requirements was explored and may be enhanced through the use of an

ontology to automatically generate a prototype of the IT infrastructure that allows the analysis of an architectural choice based on a chosen reference architecture.

Additional limitations indicate the difficulty in understanding ontologies. Understanding of the ontology may be enhanced with the use of controlled natural language interfaces that guide the stakeholders in the creation of requirements that are constrained by the vocabulary of the formal knowledge representation of the application domain and modelling language

The examination of modelling and analysis identified **enterprise modelling as an approach** to prompt further requirements elicitation. The modelling of business processes and web services were reviewed. The benefits of this approach were identified and may be extended by creating enterprise models from web services to form the basis of alignment between the business process and IT infrastructure for requirements elicitation. **Metrics for examining model recall and model understanding were also identified**.

The examination of collaboration engineering identified **an approach to the design of a recurring collaborative process model**. The benefits of such an approach were identified. The limitations of this approach were also identified and demonstrate the lack of design for virtual collaboration in global software development... The collaborative process model may be enhanced by designing the process activities for asynchronous use.

The examination of communication **identified metrics such as frequent communication to evaluate rich channels of communication to foster shared understanding**. The use of communication in global software development was reviewed with limitations around stakeholder's ability to conduct synchronous communication due to time differences. The benefits of asynchronous communication were identified Current asynchronous communication may be enhanced by providing a rich channel of communication that embeds Coughlan and Macredie's (2002) recommendations for effective communication.

In summary, the requirements that emerge from the state of the art are described in Table 2-3 and are ranked in order of priority.

Table 2-3 Requirements that emerge from the State of Art

Requirement Priority	Requirement
R1	Structure a process model around activities that include Domain Understanding, Domain Modelling, Requirements Elicitation and, Modelling and Analysis.
R2	Structure and represent application domain knowledge and model knowledge through the use of formal knowledge representation techniques to create an ontology driven system that supports ontology-driven communication.
R3	Guide the stakeholders through controlled natural language in the creation of formal knowledge representation artefacts.
R4	Create a rich channel of communication that embeds techniques for requirements elicitation such as prototyping and modelling.
R5	Generate a prototype of the IT infrastructure from the vocabulary of the ontology through developing the static part of the program in the form of type or class declarations and procedures.
R6	Use of prototype that allows the analysis of an architectural choice to identify non-functional requirements.
R7	Incorporate Coughlan and Macredie's (2002) four recommendations for effective communication in the CRESUS framework. First include business users. Second, select an adequate mix of IT architects and business users who then interact on a collaborative basis; Third, incorporate communication activities that relate to knowledge acquisition, knowledge negotiation and user acceptance. Fourth, use elicitation techniques for mediating communication for the requirements of a system such as interviews, brainstorming, prototyping and scenarios.
R8	Design a collaborative process model for asynchronous use.
R9	Create enterprise models from web services that form the basis of alignment between the business process and IT infrastructure.
R10	Separate the formal knowledge representation of the model from the diagrammatic representation.

In the next chapter, influences from the literature are used in the design of the Collaborative Requirements Elicitation Framework proposed in this thesis, named CRESUS.

3 Collaborative Requirements Elicitation Framework - CRESUS

3.1 Introduction

This chapter discusses the issues impacting on the design of a collaborative requirements elicitation framework that is comprised of a process model and support tool to address the objectives of this thesis. It describes how the state of the art from chapter two influences the design of the CRESUS process model and support tool, CRESUS-T.

The chapter continues in section 3.2.1 by introducing the CRESUS process model for fostering shared understanding that is proposed in this thesis. The process model consists of four activities: - Domain Understanding; Domain Modelling; Requirement Elicitation; and, Modelling and Analysis. This section describes the activities of the CRESUS process model including the combination of techniques collaboration, communication, prototyping, and modelling. This section also describes how shared understanding is manifested through the creation of the knowledge representation artefacts such as requirements model, IT infrastructure model and business process model.

Section 3.2.2 then introduces the support tool CRESUS-T that has been developed as part of this research. This section describes the main architectural components of the support tool that supports the CRESUS process model.

3.2 Framework Design Overview

The state of the art analysis indicates that there is a lack of research into combining collaboration, communication, prototyping and modelling techniques as part of a formalised collaborative requirements elicitation framework to foster shared understanding amongst stakeholders.

This thesis proposes such a collaborative requirements elicitation framework, comprising a process model and support tool.

3.2.1 CRESUS Process Model

The analysis of the state of the art in chapter 2 influenced the properties of the process model under design. The main properties of the process model are described in Table 3-1.The table also describes the requirements that emerged from the state of the art that influences the properties of the process model (c.f. section 2.4).

Table 3-1 Poperties of CRESUS process model

Property Number	Property	Requirement from State of the Art
P1	Provides process guidance on the order of the activities and creation of formal knowledge representation artefacts (cf. section 2.2).	R1
P2	Supports ontology-driven communication that machines can reason over and understand (cf. section 2.2.1).	R2
P3	Influences collaboration among stakeholders asynchronously (cf. section 2.3.1).	R8
P4	Supports effective communication amongst stakeholders (cf. section 2.2.1 and section 2.3.2).	R7
P5	Identifies non-functional requirements through the analysis of an architectural choice (cf. section 2.2.2).	R6
P6	Aligns the business process model and IT infrastructure model (cf. section 2.2.3).	R9

The following paragraphs outline how the CRESUS process model was designed to achieve the properties outlined above. For Property P1, process models provide guidance on the order of the activities and creation of artefacts that a project performs (Boehm, 1988). IEEE Computer Society (Bourque & Fairley, 2014) identifies four activities that influence the design of the CRESUS process model, namely requirements elicitation, requirements analysis, requirements specification and requirements validation. Van Lamsweerde (2009) distinguishes domain understanding and requirements elicitation as two process model activities. Nuseibeh and Easterbrook (2000) indicate modelling and analysing requirements as a core activity of a requirements process model. The influences from knowledge engineering indicate an additional activity of domain modelling using formalised knowledge representation techniques such as an ontological model. The influence of a formal knowledge representation to facilitate shared understanding of the application domain is represented as a distinct activity, Domain Modelling. The approach to having a distinct Domain Modelling activity is unique in comparison to traditional process models in the state of the art. Domain modelling is given more importance in the process model as it is the engine supporting ontology-driven communication which is used for creating the formal knowledge representation artefacts.

For Property P2, adapting Guarino's (1998) description[10] of an ontology-driven system to an ontology-driven process model, places the ontology as a formal knowledge representation activity in Domain Modelling that drives the remaining activities of the process model. Such an approach formalises the process model and enables ontology-driven communication among stakeholders through messages that contain expressions formulated from the vocabulary of the application domain. This approach allows machines to automatically reason over and understand the communication. .

For Property 3, collaborative engineering focuses on the need to create a process model that influences collaboration among stakeholders as opposed to just using collaborative tools. Most collaboration engineering studies have focused on face-to-face settings. However in the context of global software development, influences from the state of the art indicate that these requirement elicitation process models should be designed for asynchronous use. This indicates a need to include collaborative patterns that facilitate asynchronous use among stakeholders in each process activity.

For Property 4, the *stakeholders* are the people with an interest in the future system, which includes anyone that can influence the outcome or has knowledge of the requirements (Robertson and Robertson, 2012). According to Coughlan and Macredie (2002) for effective communication there is a requirement to include an adequate mix of IT architects and business users who then interact on a collaborative basis as the stakeholders. Hepp, Leymann, Domingue, Wahler, and Fensel (2005) discuss stakeholders such as business executives and IT architects. Requirements methodologies, such as participative design, place business users of the system at the centre of the process model. Robertson and Robertson (2013) identify a variety of stakeholders such as the sponsor, customer, and management. It is virtually impossible to include all stakeholders on a regular basis. Therefore for the purposes of this thesis a representative group of stakeholders (indicated as "process actors" in) are drawn from the roles of business executive, IT architect and business users. A *stakeholder* is anyone that has an interest in the outcome of the future system. A *business executive* may represent key

---

10 An ontology-driven system places the ontology as a component within the system. The reason for using an ontology-driven system is that it enables communication through messages that contain expressions formulated from the ontology. Guarino (1998) refers to this as ontology-driven communication. The advantage of this approach is that machines can automatically reason over and understand the communication.

decision makers such as a process owner, customer and management. An *IT architect* may represent the chief technical officer or technical staff within the IT Department. *Business users* may represent the people that will operate the system. The stakeholders communicate based on ontology-driven communication.

For Property P5, a prototype of the IT infrastructure is automatically generated based on a choice of reference architectures and vocabulary from the formal knowledge representation of the application domain. This approach allows the stakeholders analyse the architectural choice to identify non-functional requirements.

For Property P6, an organisation is modelled through the business processes and services that it provides (Greenspan & Feblowitz, 1993). Loucopoulos and Champion (1988) describe an approach to developing enterprise models using formal knowledge representation of the model. Using machine translation, alignment of the business process and IT infrastructure can be achieved by mapping the knowledge representation of the IT infrastructure to the knowledge representation of the model. The model may evolve into an executable business process model that orchestrates a prototype of the IT infrastructure deployed in a live environment. This ensures that the solution is grounded in the environment of the application domain and supports alignment of the business process and IT infrastructure.

The CRESUS process model includes activities, process actors and knowledge representation artefacts as represented in . The CRESUS process model provides step by step guidance on the order of the requirements engineering activities and the creation of artefacts. The process model consists of four activities namely: Domain Understanding, Domain Modelling, Requirements Elicitation, and Modelling and Analysis. The Requirements Elicitation activity consists of two sub activities namely Requirements Gathering, and Generation and Testing. Several artefacts are created throughout the process model. The Problem Domain Description artefact is produced as an output of the Domain Understanding activity. The Problem Domain Description describes the business process based on the requirements that surface from discussions with the Business Executive and organisational documentation. In addition, the Problem Domain Description describes additional stakeholders that have an interest in the future system. The Problem Domain Description is an input artefact to the Domain Modelling activity.

In the Domain Modelling activity, the stakeholders collaborate to identify a formal knowledge representation based on concepts and actions from the Problem Domain Description and additional knowledge they have of the application domain. Concepts are identified by nouns, and actions are identified from verbs in the Problem Domain Description. The Formal Knowledge Representation of the Application Domain artefact is produced as an output of the Domain Modelling activity. In the Requirements Elicitation activity, the Formal Knowledge Representation of the Application Domain is an input artefact that is used to constrain the vocabulary of the ontology-driven communication between stakeholders. In the Requirements Gathering sub-activity, the stakeholders create requirements through ontology-driven communication. The Requirements go through an approval process and are then stored in the Instance Model. The Instance Model, Concept Model and Actions Model form the Requirements Model. The stakeholders move into the Testing and Generation sub activity when they indicate that they have completed the Requirements Model. The IT Infrastructure Model is automatically generated from the Concepts Model, Actions Model and Instance Model. The IT infrastructure Model is presented to the stakeholders in a separate testing environment where the stakeholders can add requirements through the generated web service interfaces isolated from the actual Requirements Model and approval process. The Requirements Model and IT Infrastructure Model are output artefacts of the Requirements Elicitation activity. The Business Process Model artefact is created in the Modelling and Analysis activity. The Business Process Model is derived from the Concepts Model, Actions Model and knowledge from the stakeholders. The stakeholders create the business process elements and instance data through ontology-driven communication.

All stakeholders may participate in each CRESUS process model activity, although highlights the primary actors that are involved in each activity. The stakeholders are represented by the Business Executives such as a Process Owners that is responsible for the business process, IT Architects that build the systems and Business Users that operate the future software. The primary stakeholders are positioned in line with the process model activity that they are actively involved with. For example, the Business Executive in the form of a Process Owner is primarily involved with the Domain Understanding activity.
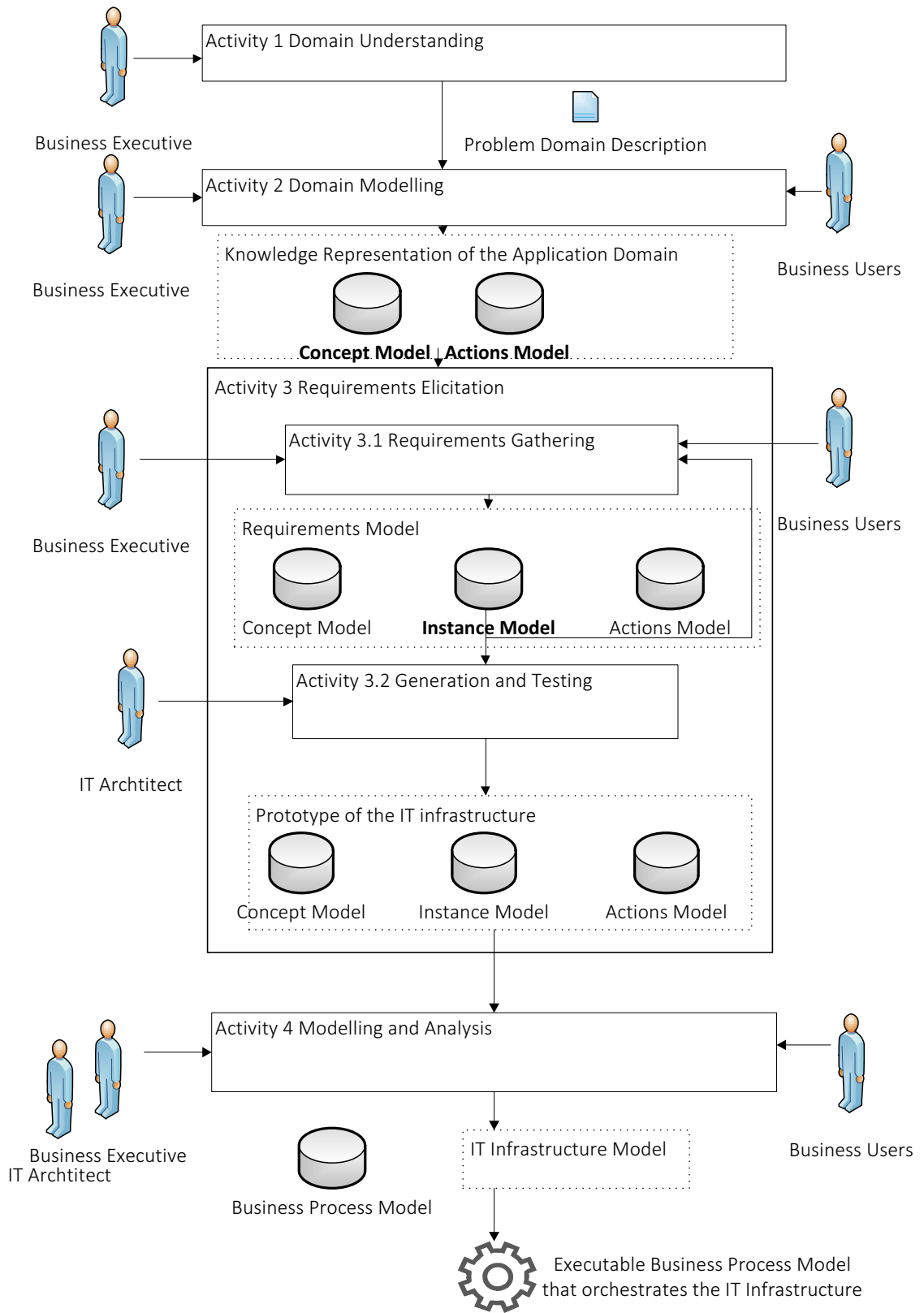
Figure 3-1 CRESUS Process Model

A detailed description of each activity *using collaborative patterns* (cf. section 2.3.1) is discussed in sections 3.2.1.1 through to section 3.2.1.4.

3.2.1.1 Activity 1 Domain Understanding

Van Lamsweerde (2009) indicates that this activity consists of studying the current system within its organisational and technical context. The aim of this activity is to attain an understanding of the application domain based on the organisation, scope of the current system, stakeholders, and the strengths, and weakness of the current system (Van Lamsweerde, 2009 p. 30).

The organisation is described by its goals, strategy, objectives, business policies, roles played by the organisational units, actors and dependencies between them (Van Lamsweerde, 2009 p. 30). A *business goal* is a broad primary outcome. A *strategy* is an approach to achieve the goal. An *objective* is a measurable step that an organisation takes to make its strategy succeed. A *business policy* is a directive that is not directly enforceable whose purpose is to govern or guide the enterprise (Object Management Group, 2004). Business policies must be interpreted and refined to turn them into business rules which provide specific, practicable guidance to implement business policies (Object Management Group, 2004 p. 15). In essence business rules are lists of statements that direct what can or cannot be done. Business rules also provide conditions for making a decision. Business requirements enable the implementation of and compliance with business rules (Lam, 2006). Business rules define and constrain business processes ensuring that business activities are aligned with business goals and strategies. An example of a business policy such as "Safety is our first concern" may be refined into a business rule "A hard hat must be worn in a construction site". "A foreman inspects the attire of personnel on the site" is a business requirement that enables the implementation and compliance with the business rule.

Robertson and Robertson (2013) describe the scope of the current system as the organisational area that is affected by the system. The scope of the current system is described by the systems objectives, components forming it, concepts on which it relies, tasks involved, information flowing through it, constraints and regulations to which the system is subject.

For effective collaboration and communication, the stakeholders include an adequate mix of IT and business users. Hepp Leymann Domingue Wahler and Fensel (2005) discuss stakeholders such as business executives and IT architects. Requirements methodologies such as *participative design* place business users of the system at the

centre of the process model. For the purposes of this thesis stakeholders represent business executives, IT architects and business users. Robertson and Robertson (2013) identify a variety of stakeholders such as the sponsor, customer, and management, which for simplicity will be categorised as business executives.

The strengths and weakness of the current system is based on the perception of the identified stakeholders (Van Lamsweerde, 2009, p. 31).

An understanding of the application domain typically involves studying key documents, and interviewing stakeholders. The *structured interview* is a primary requirement elicitation technique that is useful for getting an understanding of the application domain. An interview takes place with the business executive such as the process owner that is responsible for the business process, and the interviewer as represented by a requirements engineer. The interviewer asks the business executive questions and records answers. The interviewer creates a *problem domain description* that describes the business process of the application domain from the interview transcripts. The problem domain description is refined and validated by the business executive. The collaborative patterns are described in Table 3-2.

Table 3-2 Identifying Collaborative Patterns for Domain Understanding

Requirements Engineering Activity	Collaborative Tasks	Artefact
Domain Understanding	Requirements Engineer gathers key documents that describe the organisation, scope of the current system, stakeholders and the strengths and weakness of the current system.	Problem Domain Description
	Business Executive generates individual understanding of the business process.	
	Requirements Engineer asks questions for clarification	
	Business Executive identifies who is responsible for approving the requirements and identifies the stakeholders involved in requirements elicitation.	
	Business Executive evaluates the Problem	

	Domain Description	

The collaborative process model that supports the Domain Understanding activity is shown in Figure 3-2.
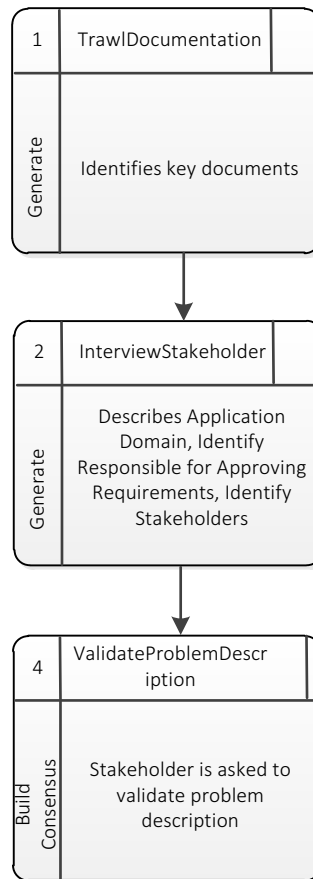
Figure 3-2 Collaborative Process for Domain Understanding Activity

The collaborative methods used in the Domain Understanding activity are the trawl of documentation, interview stakeholder and validate the problem domain description. *Trawl of documentation* is used for identifying concepts and actions such as relationships from the application domain. *Interviewing* the stakeholder is used for identifying a problem description of the application domain. The *Build Consensus* collaborative pattern allows the requirements engineer to continuously track the level of consensus with the stakeholders regarding the description of the application domain (Kolfschoten, Briggs, Appelman, & de Vreede, 2004).

3.2.1.2 Activity 2 Domain Modelling

The second activity of the CRESUS framework involves acquiring and modelling knowledge from the application domain in such a manner to constrain the natural

language communication and allow for semantic interpretation by machines. Influences from the state of the art (cf. section 2.2.1) indicate a need for the use of a formalised knowledge representation that is amenable by machines to automatically generate the IT infrastructure through developing the static part of the program in the form of type or class declarations and procedures.

The formalised knowledge representation is described by the concepts and actions in the form of relationships derived from stakeholders using the brainstorming technique. *Brainstorming* is a process whereby stakeholders engage in informal discussion to rapidly generate as many concepts and actions as possible without focusing on any one in particular. One of the advantages in using brainstorming is that it promotes free thinking and expression. The stakeholders use the problem domain description as input to this task. The identified concepts and actions form part of the knowledge representation model of the application domain. For example, *Lecturer teaches Class* represents concepts Lecturer and Class. The relationship teaches represents an action. The conceptual model and actions model is validated by all stakeholders. The collaborative tasks are described in Table 3-3. The business executive and business users are the key stakeholders during this process activity.

Table 3-3 Identifying Collaborative Patterns For Domain Modelling

Requirements Engineering Activity	Collaborative Tasks	Artefact
Domain Modelling	Stakeholders generate concepts and actions in the form of relationships from description of the application domain	Ontology Model
	Stakeholders validate the formal knowledge representation model	

The collaborative process that supports the Domain Modelling activity is shown in Figure 3-3.
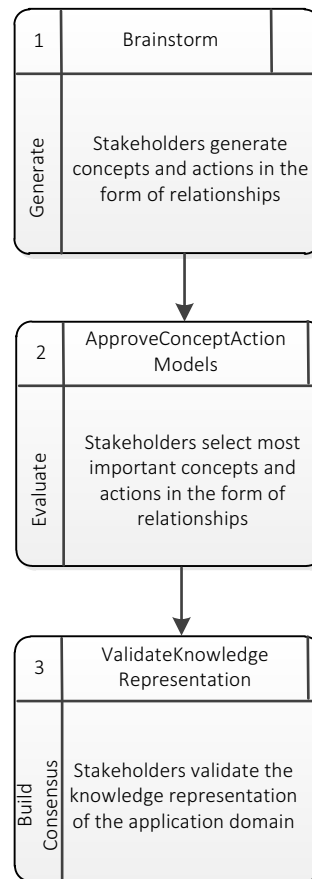
Figure 3-3 Collaborative Process for Domain Modelling Activity

The *Generate* collaborative pattern is used to generate a broad, diverse set of highly creative concepts and relationships contributed by the stakeholders (De Vreede, Kolfschoten, & Briggs, 2006). The *Evaluate* collaborative pattern allows a group to evaluate a number of concepts and relationships with respect to a single criterion such as relevancy to the problem description (Kolfschoten, Briggs, Appelman, & de Vreede, 2004). The stakeholders have to select the most important concepts and relationships supported by a voting mechanism. The *Build Consensus* collaborative pattern allows the requirements engineer to continuously track the level of consensus within the group regarding the ontology model of the application domain.

3.2.1.3 Activity 3 Requirement Elicitation

Requirements elicitation involves discovering requirements that will shape the future system as they emerge from domain understanding (Van Lamsweerde, 2009). Requirements are elicited from sources such as the business goals, strategy, objectives, domain knowledge, stakeholders, business rules, operational environment and organisational structure (Bourque & Fairley, 2014). Influences from the state of the art

54

indicate a need to combine elicitation techniques such as the prototype to get an understanding of the architectural choice on the IT infrastructure that supports the business needs.

Stakeholders prefer to communicate requirements through natural language. Using ontology-driven communication, the stakeholders are prompted in the construction of the requirements. The vocabulary for constructing the requirements is taken from the content words of the underlying formal knowledge representation of the application domain.

Requirements have to be elicited using techniques, such as prototype, for getting the stakeholders to articulate requirements. The prototype provides the stakeholders with a method to identify non-functional requirements. Using machine translation, a prototype of the IT infrastructure is automatically generated based on the static part of the architectural components in the form of type or class declarations, procedures and database schema derived from the knowledge representation of the underlying application domain and reference architecture. The deployed prototype represents a mechanism for the stakeholders to analyse the architectural choice. The stakeholders can test the architectural components, provide information through the interfaces, and view the databases directly. The collaborative tasks are described in Table 3-4.

Table 3-4 Identifying Collaborative Patterns for Requirements Elicitation

Requirements Engineering Activity	Requirements Engineering Sub Activity	Collaborative Tasks	Artefact
Requirements Elicitation	Requirements Gathering	Stakeholders communicate requirements	Requirements Model
		Stakeholders validate the requirements	
		Stakeholders indicate that they have completed the sub activity	
	Generation and Testing	IT architect automatically generates a prototype of the IT infrastructure	IT Infrastructure Model
		The stakeholders test the prototype	
		Stakeholders indicate that they have completed the sub activity	

The *Generate* collaborative pattern allows the stakeholders communicate using ontology-driven communication for representing the requirements which are constrained by the language of the application domain. The *Evaluate* collaborative pattern allows the stakeholders approve the requirements. The *Build Consensus* collaborative pattern allows the stakeholders validate the requirements model. The second *Generate* collaborative pattern allows the stakeholders automatically create, deploy and test a prototype of the IT infrastructure. The second *Build Consensus* collaborative pattern allows the stakeholders validate the prototype of the IT infrastructure. The collaborative process that supports the Requirements Elicitation activity is shown in

Figure 3-4.

1 | SemanticCommunication

Generate

Stakeholders communicate requirements constrained by the vocabulary of the problem domain

2 | ApproveRequirements

Evaluate

Stakeholders approve the requirements

3 | ValidateRequirementsModel

Build Consensus

Stakeholders complete requirements gathering sub activity

4 | PrototypeOfITInfrastructure

Generate

Automatically generate, deploy and test a prototype of the IT infrastructure

5 | ValidateITInfrastructure

Build Consensus

Stakeholders complete generation and testing sub activity

Figure 3-4 Collaborative Process for Requirements Elicitation Activity

3.2.1.4 Activity 4 Modelling and Analysis

The IEEE body of knowledge indicates that requirement analysis is concerned with analysing requirements to detect and resolve conflicts between requirements; discover the bounds of the software and how it must interact with its organisational and operational environment; and elaborate system requirements to derive software

requirements. Conceptual modelling of the concepts, actions and dependencies of the application domain in the form of a *scenario,* both aids understanding of the problem, and depicting a solution (Bourque & Fairley, 2014 p. 1-8). Scenarios are common in modelling software and as an elicitation technique they provide context to stakeholder requirements. There are many types of models to choose from in requirements analysis such as data flow models, state models and user interaction models. The nature of the problem influences the choice of the modelling notation adopted. Influences from the state of the art (cf. section 2.2.3) indicate a need for creating enterprise models from web services to form the basis of aligning the business process with the IT infrastructure.

As this thesis is focused on the alignment of the business process and IT infrastructure then it is natural to adopt a modelling notation that supports the analysis of the business process. Models can be used to depict scenarios where the boundary separates the business users or systems in the external environment.

Machine translation of the formal knowledge representation of the application domain can be used to create the model ontology. This is achieved by automatically mapping the architectural components and procedures of the prototype as described in the concepts and actions model to the equivalent elements in the model ontology. Using ontology-driven communication, the stakeholders communicate to create a formal knowledge representation of the model. Through machine translation the underlying model may be represented graphically in diagrammatic form. The formalised knowledge representation of the model may be deployed as an executable model in the IT infrastructure. The collaborative tasks are described in Table 3-5.

Table 3-5 Identifying Collaborative Patterns for Modelling and Analysis

Requirements Engineering Activity	Collaborative Tasks	Artefact
Modelling and Analysis	Automatically generate a model of the business process from the formal knowledge representation of the application domain	Business Process Model
	Stakeholders communicate requirements of the model linguistically	
	Model is represented by a business process diagram	
	Stakeholders indicate that they have completed the activity	
	Business process is mapped to an executable model and deployed	Executable Business Process Model

The collaborative process that supports the Modelling and Analysis activity is shown in Figure 3-5.

The *Generate* collaborative pattern, "Modelling", is a collaborative method that creates a formal knowledge representation of the business process model that is aligned to the IT infrastructure.

The *Generate* collaborative pattern "Ontology-Driven Communication" allows stakeholders communicate requirements that are constrained by the formal knowledge representation of the model. The *Generate* collaborative pattern "Model Representation" displays a diagram that represents the formal knowledge representation of the model. The *Build Consensus* collaborative pattern allows the stakeholders validate the model. The *Generate* collaborative pattern allows the stakeholders create an executable model that is deployed in the IT infrastructure.
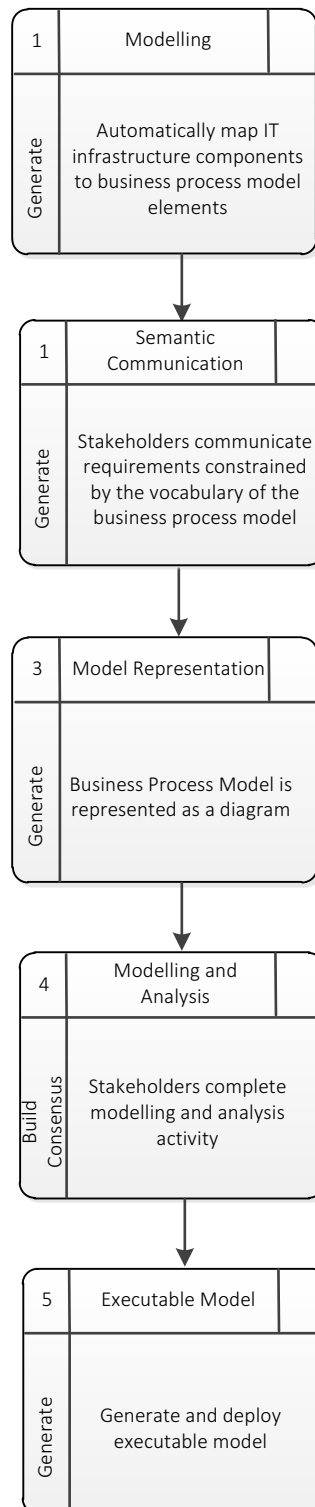
Figure 3-5 Collaborative Process for Modelling and Analysis Activity

3.2.1.5 Summary

Section 3.2.1 described the properties for the design of a formalised collaborative requirements elicitation process model, CRESUS. The CRESUS process model comprises of four activities: - Domain Understanding; Domain Modelling; Requirements

Elicitation; and, Modelling and Analysis. A representative group of stakeholders are drawn from the roles of business executive, IT architect and business users and represent anyone that has an interest in the outcome of the future system. The CRESUS process model influences collaboration among stakeholders through the use of collaborative patterns. Stakeholders collaborate, communicate and model to create knowledge representation artefacts such as requirements model and business process model. Through an automatic code generation process, a prototype of the IT infrastructure is automatically created based on a choice of reference architecture and vocabulary from a formal knowledge representation of the application domain.

3.2.2 Support Tool, CRESUS-T

The architecture of the support tool CRESUS-T includes a communication mechanism that is constrained by the underlying formal knowledge representation of the application domain and business process model, and a machine translation component that automatically generates a prototype of the IT infrastructure and maps the IT infrastructure components to business process model elements. This is represented in Figure 3-6. The analysis of the state of the art in chapter 2 influenced the properties of the support tool under design. CRESUS-T support tool has been designed to support the properties described in Table 3-6 The table also describes the requirements that emerge from the state of the art that influences the properties of the support tool, CRESUS-T (c.f. section 2.4).

Table 3-6 Properties of CRESUS-T Support Tool

Property Number	Property	Requirements from State of the Art
P1	Guides the stakeholders on the order of the activities and the creation of the artefacts in the CRESUS process model (cf. section 2.3.1).	R1 and R2
P2	Guides the stakeholders in the creation of requirements (cf. section 2.2.1).	R3
P3	Provides a rich channel for effective communication (cf. section 2.3.2).	R2, R4 and R7

P4	Provides machine translation to automatically generate a prototype of the IT infrastructure (cf. section 2.2.1).	R2 and R5
P5	Provides alignment of the business process and IT infrastructure (cf. section 2.2.3).	R9
P6	Provides machine translation to automatically generate a diagrammatic representation of the model (cf. section 2.2.1).	R10

The following paragraphs outline how the support tool, CRESUS-T was designed to achieve the properties. For Property P1, the first two activities of the CRESUS process model namely Domain Understanding and Domain Modelling focus on the current system that is in place and helps to build up background context for the future system for which requirements are being elicited. CRESUS-T was designed to support the central two activities of the CRESUS process model, namely Requirements Elicitation and, Modelling and Analysis.

For Property P2, a controlled natural language interface allows the system prompt the stakeholders on the construction of sentences that relate to the requirements using natural language. The sentences are based on a formal knowledge representation of the application domain and modelling language. This approach facilitates constraining the vocabulary of the natural language and allows ontology-driven communication. In addition, this approach is amenable to machine understanding and automatic machine translation.

For Property P3, asynchronous communication is the predominant form of communication in global software development. The rich channel of communication integrates Coughlan & Macredie (2002) techniques for mediating the communication of requirements as follows:-

- A web based tool that allows the stakeholders to collaborate asynchronously.
- The incorporation of communication activities that centre around the creation of artefacts such as the requirements model, IT infrastructure model, and business process model that facilitate knowledge acquisition, knowledge negotiation and user acceptance.

- The use of elicitation techniques for mediating communication for the requirements of a system such as prototyping, modelling and scenarios.

For Property P4, machine translation automatically generates the IT infrastructure based on the concepts and actions in the form of relationships from the underlying formal knowledge representation of the application domain. The resultant nouns and verbs are applied to a choice of reference architectures to create a prototype consisting of architectural components such as software programs and associated data models. Deployment of the architectural components results in the creation of the prototype platform that represents the evolution of the IT infrastructure. Stakeholders can retrieve, create, delete and modify data through the software program interfaces.

For Property P5, using machine translation, alignment of the business process and IT infrastructure can be achieved by mapping the knowledge representation of the IT infrastructure to the knowledge representation of the business process model. Thereafter the knowledge representation of the business process model can be linguistically augmented through ontology-driven communication amongst the stakeholders as they model a business process scenario.

For Property P6, through machine translation the underlying knowledge representation of the business process model can be represented in diagrammatic form that is dependent on the model notation. Such an approach allows stakeholders and especially business executives who may be unfamiliar with modelling notations to still understand and validate the requirements based on natural language.

The knowledge representation artefacts that are created from the CRESUS-T support tool are the Requirements Model, IT Infrastructure Model and Business Process Model. The formal knowledge representation artefacts represent a manifestation of the stakeholders shared understanding. In order to achieve these artefacts, the CRESUS-T was built around a Communication Mechanism, Prototype of IT Infrastructure Transformation component, Model Alignment component and a Diagram Renderer component as described in Figure 3-6.
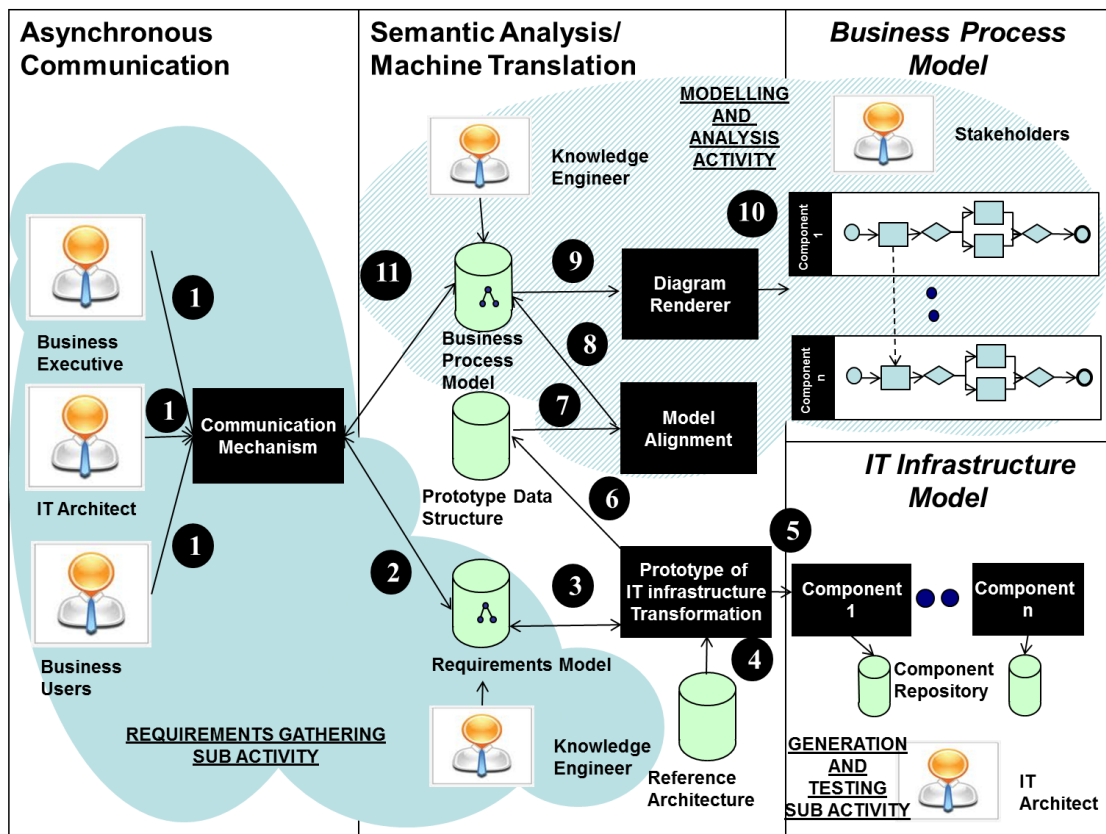
Figure 3-6 Architecture of the Support Tool, CRESUS-T

The Communication Mechanism facilitates ontology-driven natural language communication between the Business Executive, IT Architect and Business Users (see message 1) in Figure 3-6. This is further described in section 3.2.2.1. The stakeholders can collaboratively communicate and validate requirements that are stored in the Requirements Model (see message 2). The Prototype of IT Infrastructure Transformation component automatically generates a prototype of the IT infrastructure (see message 5) based on the vocabulary of the application domain (see message 3) and a reference architecture (see message 4). The components. operations and parameters of the generated IT infrastructure are stored in the Prototype Data Structure database (see message 6). The prototype of the IT infrastructure transformation component is further described in section 3.2.2.2.

The model alignment component aligns the components of the IT infrastructure as described in the Prototype Data Structure database (see message 7) to the equivalent elements in the business process model (see message 8). The model alignment component is further described in section 3.2.2.3.

64

The Diagram Renderer component transforms the formal knowledge representation of the business process model (see message 9) to a diagrammatic representation (see message 10). This is further described in section 3.2.2.4. The stakeholders may model a scenario of the business process that is aligned to the IT infrastructure (see message 11).

3.2.2.1 Communication Mechanism

The communicaiton mechanism is responsible for handling the communication between the stakeholders and the underlying formal knowledge representation of the application domain. Using ontology-driven communication, the stakeholders are prompted in the construction of the requirements. The vocabulary for constructing the requirements is taken from the content words of the formal knowledge representation of the underlying application domain. The approach to verbalise the formal knowledge representation into natural language is shown in

Table 3-7 (Kaljurand and Fuchs, 2007). The formal knowledge representation concepts of Object, Class and Property are mapped to nouns and verbs.

Table 3-7 Verbalising the Formal Knowledge Representation in Natural Language

Corresponding verb and noun phrases	Knowledge Representation of the Application Domain	Example
Common noun	Named Class	Lecturer
Proper noun	Named Object	JohnSmith
Transitive verb	Named Property	Teaches

The grammar of the language defines and constrains the form and meaning of the sentences. The sentences have the following structure:-

Subject + verb + complement (Fuchs, Kaljurand, Kuhn, 2008)

Where subject and complement are represented by nouns. In a course registration system in a third level educational institution for example, this would be similar to *John Smith teaches Software Development* with the semantic meaning of *Lecturer teaches Module*. The resultant requirements are validated and form part of the requirements model.

3.2.2.2 Prototype of the IT Infrastructure Transformation

Reference architectures represent models of application domain software structures (Berzins, Martell, Luqi, & Adams, 2008). They provide a template solution for the IT infrastructure. The code generation process described in Figure 3-7 uses the reference architecture and sets of rules for tailoring the prototype of the IT infrastructure based on extending Abbots (1983) and Booch's (1986) textual analysis technique (cf. section 2.2.1) to the vocabulary from the formal knowledge representation of the application domain (Berzins, Martell, Luqi, & Adams, 2008).
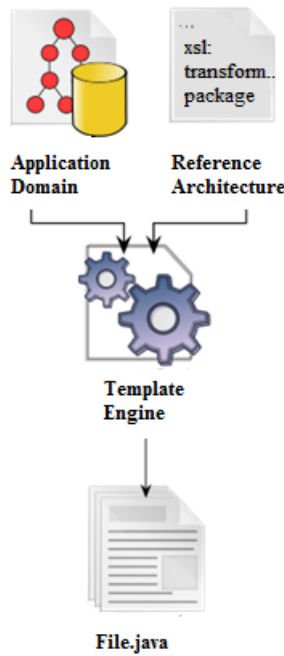
Figure 3-7 Code Generation Process

Table 3-8 describes how to derive classes (or web services) from common nouns, objects (or web parameters) from proper nouns and operations (or web methods) from verbs arising out of the vocabulary from the formal knowledge representation.

Table 3-8 Identifying Candidate Software Programmes

Vocabulary	Booch (1986) Object Orientation	Service Oriented Architecture
Common noun	Class	Web Service
Proper noun	Object	Web Parameter
Transitive verb	Operation	Web Method

A template engine combines the application domain ontology with a reference architecture in order to automatically generate the source code of the prototype as described in Figure 3-7. This prototype represents the IT infrastructure and when deployed represents a mechanism for the stakeholders to analyse the architectural choice. The stakeholders can test the architectural components, provide information through the interfaces, and view the databases directly.

3.2.2.3 Model Alignment

The purpose of the business process model is to describe the behaviour of the organisation in which that system will operate and display significant features and characteristics of future systems. Using machine translation, the verb and noun phrases of the formal knowledge representation of the application domain that represents the IT infrastructure elements are aligned to a formal representation of the business process model elements through mapping as described in Table 3-9.

Table 3-9 Mapping from Application Domain to Model

Corresponding verb and noun phrases	Formal Knowledge Representation of the Application Domain	IT Infrastructure elements e.g. Service Oriented Architecture	Business Process Model elements
Common noun	Named Class	Web Service	Participant
Transitive verb	Named Property	Method	Task participant performs

The stakeholders can build the model linguistically through ontology-driven communication that uses the underlying model ontology.

3.2.2.4 Diagram Renderer

The diagram renderer is responsible for representing the formal knowledge representation of the business process model in diagrammatic form using machine translation. This approach ensures separation of concerns between the model and the representation of the model.

3.3 Summary

Chapter 0 discussed the design of the CRESUS framework that consists of the CRESUS process model and tool support, CRESUS-T. The properties of the process model and support tool were defined.

The properties of the CRESUS process model are based on the influences emerging from the analysis conducted in chapter 2. The CRESUS process model provides step by step guidance on the order of the requirements engineering activities and formal knowledge

representation artefacts created. The process model incorporates patterns of recurring tasks to influence collaboration among stakeholders.

The properties of the CRESUS-T support tool are technical in nature and based on the influences emerging from the analysis conducted in chapter 2. A proposed architecture for the support tool CRESUS-T was presented. The CRESUS-T support tool was designed to be used asynchronously where the stakeholders can collaboratively communicate and validate requirements, automatically generate a prototype of the IT infrastructure and model a scenario of the business process that is aligned to the IT infrastructure.

The designed framework combined collaboration, communication, prototyping and modelling in order to foster shared understanding amongst stakeholders and this claim will be evaluated in chapter 5.

Following on, Chapter 4 describes the implementation of the collaborative requirement elicitation framework design that is outlined in this chapter.

4 Implementation

4.1 Introduction

This chapter describes the implementation of the collaborative requirement elicitation framework described in the design chapter. CRESUS is presented as a framework that comprises of a process model with four activities and implementation of a support tool CRESUS-T that supports the two central activities of the process model.

This chapter describes how the CRESUS process model was concretely applied in a case study to the manual business process for the roll out of the IT software image to desktop computers in laboratories at the National College of Ireland.

The chapter also details the technological architecture of the CRESUS-T support tool including various aspects of the implementation such as: the representation of formal knowledge; the support for controlled natural language communication; the automatic generation of the IT infrastructure; and the support for modelling and representing the business process.

4.2 Implementation of the CRESUS Process Model

In order to evaluate the proposed CRESUS framework, it was decided to apply it in a real-world situation in the organisation in which the author of this thesis is employed, the National College of Ireland (NCI). This decision was prompted by the desire to demonstrate the contribution of the proposed framework to solving a real problem, despite the significant challenges that would be imposed by the setting. In this section, the real-world setting that was chosen for focus is described.

The real-world setting chosen was the rollout of the IT software image business process within NCI. The business process owner for the roll out of the IT software image is a senior administrator in the IT Department at NCI. Currently there is no documentation around the roll out of the IT software image business process. As part of the *Domain Understanding* activity of the CRESUS process model, the Senior IT administrator was interviewed in February 2009 in NCI. A problem domain description of the business process was created based on the interview. The problem domain description was validated by the Senior IT administrator. This activity was carried out synchronously. The stakeholders for the business process were identified as course directors and

lecturers in the School of Computing. The lecturers were responsible for the delivery of the module. The module comprised of lectures and tutorials, and may require desktop computers. The course director is responsible for the programmes that the modules are part of.

As part of the *Domain Modelling* activity of the CRESUS process model, the formal knowledge representation of the application domain was implemented as an ontology during a brainstorming session in February 2009. The brainstorming session consisted of the course director, IT administrator and lecturer that represented the stakeholder roles of business executive, IT architect and business user. The stakeholders used the problem domain description from the Domain Understanding activity of the CRESUS process model to generate a list of concepts, actions in the form of relationships, rules and instance data from the application domain. The stakeholders identified a sample of concepts such as *Lecturer*, *Module*, *Software*, and *SoftwareVersion*, in addition to a sample of relationships such as *teach*, *require* and *hasVersion*. The ontology was arranged by the author of this thesis as "subject predicate object" triples such as *Lecturer teaches Module*; *Module requires Software*; and *Software hasVersion SoftwareVersion*. The stakeholders identified rules such as "If a lab hosts Computer, and Computers host Software -> Lab hosts Software". The stakeholders identified a sample of instance data such as "PaulStynes teaches WebDevelopment". The stakeholders identified a rule system around approving requirements based on *Business User enter requirements*; *Business Executive approves the requirements*; *IT Architect approves the requirements*. The knowledge engineer (author of thesis) created the application domain ontology based on the identified concepts and actions in the form of relationships. The ontology was modified and validated with the stakeholders in a validation meeting in February 2009. This activity was carried out synchronously.

The *Requirements Elicitation* activity of the CRESUS process model was implemented asynchronously using the support tool, CRESUS-T. The application domain ontology model was imported into CRESUS-T. The natural language of the application domain was constrained based on the vocabulary defined in the ontology. As part of the *Requirements Gathering* sub-activity of the CRESUS process model, the stakeholders were guided in the construction of the requirements based on the "subject predicate object" triple. The *business executive* approved the requirements that were entered by the *business user*. The *IT architect* then approved the requirements that they could

implement. The approved requirements formed part of the *requirements model*. The stakeholders moved into the *Generation and Testing* sub-activity of the CRESUS process model, when they indicated through a *Build Consensus* collaboration pattern, that they had completed the *Requirements Gathering* sub-activity in CRESUS-T. In the Generation and Testing sub-activity, a prototype of the IT infrastructure was automatically generated based on the requirements model and reference architecture. The *reference architecture* components were represented by web services ("Web Services Activity Statement," n.d.) and the *data model* was represented by XML documents stored in an eXist database ("eXistdb - The Open Source Native XML Database," n.d.; "eXist-db v1.4.2," n.d.). A WSDL interface (Christensen, Curbera, Meredith, & Weerawarana, 2001) was presented to the stakeholders where they could add instance data through web method calls.

The *Modelling and Analysis* activity of the CRESUS process model was implemented asynchronously using the support tool, CRESUS-T. The formal knowledge representation of the model was based on an ontology (Rospocher, Ghidini, & Serafini, 2009) of the *Business Process Modelling and Notation* (BPMN) (OMG, 2011). The common nouns from the application domain ontology that represented the names of the Web Services that formed part of the IT infrastructure component were mapped to the ontology class "Participants" in the BPMN ontology. A "Participant" can be a specific partner entity (such as a company) or can be a more general partner role (such as a lecturer, buyer, or seller) (OMG, 2011 p. 114). Participants can be realised by a service such as a Web Service (OMG, 2011 p. 116). The stakeholders can then create a business process model using ontology-driven communication based on the BPMN ontology. Through machine translation the BPMN ontology is automatically represented in a diagrammatic form.

The application of the CRESUS process model to the roll out of the IT software image business process formed the basis for the evaluations in chapter 5.

4.3   Technological Architecture View of CRESUS-T

CRESUS-T is a web based application that is implemented in Java ("Java Platform, Standard Edition Development Kit (JDK) v7.5," n.d.) using the echo web framework ("Echo Web Framework v2.2.1," 2009). The web based application was hosted on the Amazon Web Service ("Amazon Web Services," n.d.) at http://54.238.49.81:8081. The

architecture of CRESUS-T is illustrated in Figure 4-1. The architecture facilitates the collaboration, communication, prototyping and modelling of requirements in the alignment of the business process and IT infrastructure. CRESUS-T achieves this by implementing open source components and Application Platform Interfaces.
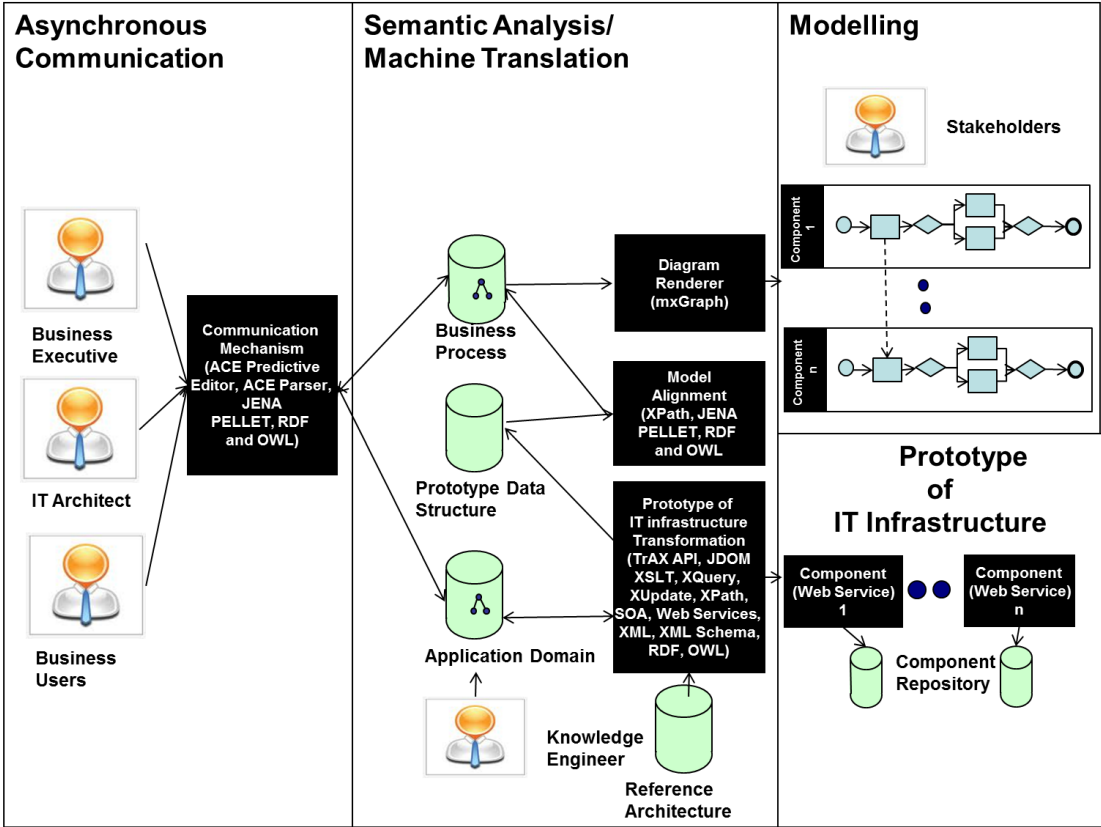
Figure 4-1 CRESUS-T Architecture

4.3.1 Controlled Natural Language Communication

Ontology-driven communication allows the stakeholders communicate requirements linguistically based on an underlying domain and model ontology to facilitate human and machine understanding. Using ontology-driven communication, the stakeholders communicate through the ACE editor ("ACE Editor," 2011) as shown in Figure 4-2. Attempto Controlled English Parsing Engine (APE) (Kaljurand and Fuchs, 2006; Kaljurand, 2013) ensures that the grammar for the requirements is based on a subset of natural language, namely Attempto Controlled English (Fuchs, Kaljurand, and Kuhn, 2008). This results in machine accessible semantics that are automatically processable and amenable to machine translation. In addition, the parsing engine converts the sentences created in ACE editor into RDF triples. The APE is deployed on port 8000. The ACE Editor and APE are 3rd party products in the CRESUS-T architecture.

The ACE editor is a predictive authoring tool that guides the stakeholders during the construction of the sentences. The ACE editor source code is part of ACEWiki ("AceWiki v0.5.1," 2011; Kuhn, 2008; Kuhn, 2008a). ACE Editor is a controlled natural language interface and is written in Java based on the echo 2 web framework. The tool ensures syntactic, lexical and semantic correctness of the sentences that describe the requirements. The authoring tool enforces syntactic correctness to ACE text in addition to a subset of ACE that supports the limited expressiveness of OWL. In ACE, the terminology is represented as a lexicon. ACE editor ensures lexical correctness by forcing the stakeholders to use only words that are in the lexicon. These words consisting of nouns and verbs are extracted from the application domain ontology and BPMN ontology. The tool ensures semantic correctness through the use of the underlying application domain and BPMN ontologies that guides the construction of the sentences in the form of subject verb complement.
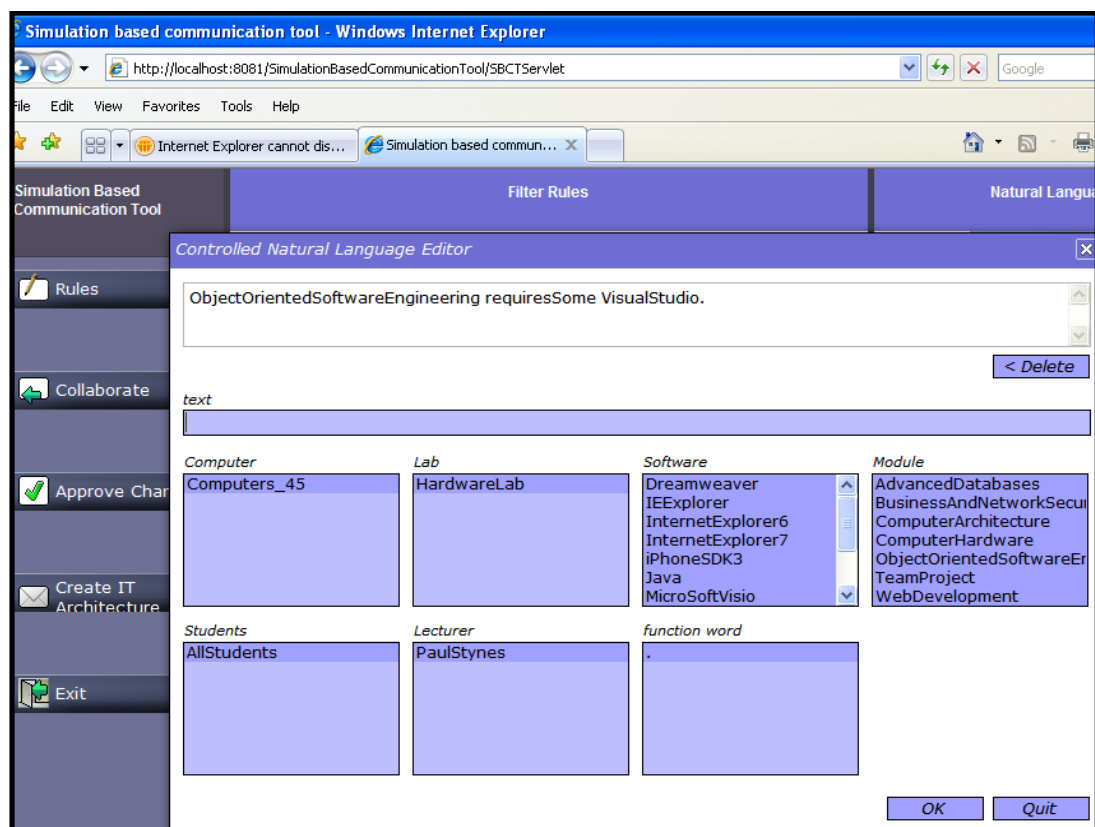
Figure 4-2 Controlled Natural Language Editor

4.3.2 Application Domain Ontology

Formal knowledge representation techniques such as ontologies are created to formalise the CRESUS process model. The application domain ontology is initially created using

an external editor, protégé ("Protégé v4.2," 2013). The ontology is imported into the server side implementation of CRESUS-T and stored as a Jena ontological model ("Apache Jena v2.6.4," 2010). Jena is an open source semantic web framework for building Java applications. Jena provides support to create and read RDF and OWL. RDF is a standard model for data interchange on the Web. It is a framework for expressing information about resources such as documents, people, physical objects and abstract concepts. It allows for statements to be made about resources based on <subject> <predicate> <object>. An RDF statement expresses a relationship between two resources where the subject and the object represent the two resources being related and the predicate represents the nature of their relationship. OWL is a semantic web language designed to create an ontology that represents knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language. Knowledge expressed in OWL can be reasoned over automatically by machines. The Jena framework uses the Pellet reasoner ("Pellet: OWL 2 Reasoner for Java v2.3.0," 2011) to reason over the OWL application domain ontology.

4.3.3 Business Process Model and Notation Ontology

The Business Process Model and Notation (BPMN) is an international standard language for business process modelling. One of the drivers for BPMN is to create a simple and understandable mechanism for creating business process models by all business users (Object Management Group, 2011). The graphical elements of the notion are organised into five basic categories as described in Figure 4-3. These elements are flow objects; connecting objects; swimlanes; artefacts; and data objects.

Core Set of BPMN Elements
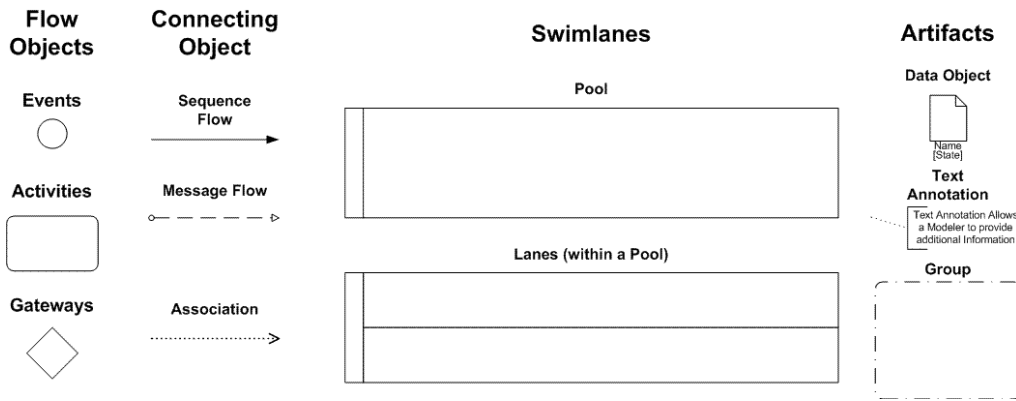
Figure 4-3 BPMN Elements

Flow objects define the behaviour of a business process. Flow objects consist of events, activities and gateways. There are three types of events that affect the flow namely, start, intermediate, and end. An activity is a generic term for work that an organisation performs in a process of which there are two types, namely sub-process, and task. A gateway is used to control the divergence and convergence of sequence flows in a process through branching, forking, merging, and joining. Sequence flows, message flows, associations and data associations are the four ways to connect flow objects. A sequence flow shows the order that activities will be performed in a process. A message flow is used to show the flow of messages between two pools as represented by participants. A message represents the contents of a communication between two participants. An association is used to link information and artefacts with BPMN graphical elements. Swimlanes are a grouping mechanism for the primary modelling elements pool and lanes within a pool. A pool is the graphical representation of a participant in collaboration. A lane is a sub-partition within a process, sometimes within a pool and is used to organize and categorize activities. Artefacts provide additional information about the process of which there are three namely, text annotation, group, and data object. Text Annotations provide additional text information for the reader of a BPMN diagram. Text annotations and other artefacts can be associated with the graphical elements. A group is a grouping of graphical elements that are within the same category. Data objects provide information about what activities require to be performed and/or what they produce. (Object Management Group, 2011).

The formal knowledge representation of BPMN was implemented using BPMN ontology (Rospocher Ghidini and Serafini, 2014; Rospocher & Serafini, 2009). BPMN also

supports the generation of executable processes through a mapping of BPMN and BPEL ("Web Services Business Process Execution Language v2.0," 2007). The ontology is imported into the server side implementation of CRESUS-T and stored as a Jena ontological model ("Apache Jena v2.6.4," 2010) that uses the Pellet ("Pellet: OWL 2 Reasoner v2.3.0," 2011) reasoner to reason over the BPMN ontology.

4.3.4 Reference Architecture

The reference architecture is a set of template instructions for transforming XML into Java source code. The reference architecture is based on an XSLT stylesheet (Saxonica, 2007). Extensible Stylesheet Language (XSL) (Berglund, 2006) is a style sheet language for XML documents. XSL Transformations (XSLT) provides the ability to transform an XML document into other formats such as Java source code representing a prototype of the IT infrastructure. A sample code snippet of the reference architecture is illustrated in Figure 4-4. The code snippet contains the instructions for creating the software programs that represent the prototype of the IT infrastructure.

```
 <xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output encoding="UTF-8" method="xml"/>
<xsl:param name="vClassName"/>
<xsl:template match="/">              <!-- DOCUMENT ROOT -->
        package test;
        import javax.jws.Oneway;
        import ….
  <xsl:apply-templates/>
 </xsl:template>
 <xsl:template match="*">          <!--ALL ELEMENTS -->
     <xsl:apply-templates/>
 </xsl:template>
 <xsl:template match="text()">       <!-- ALL TEXT NODES -->
     <xsl:apply-templates/>
 </xsl:template>
<xsl:template match="Class">
        <xsl:if test="ClassName=$vClassName">
        @WebService()
        public class <xsl:value-of select="./ClassName"/>{
        private final static String URI = "xmldb:exist://54.238.49.81:8680/exist/xmlrpc";
        private final static String DRIVER = "org.exist.xmldb.DatabaseImpl";
        private final static String collectionSim = "/db/SBCT/Sim";
```

```
        …
     @WebMethod(operationName = "getAll<xsl:value-of select="ClassName"/>Names")
     public                    java.util.ArrayList<xsl:text                    disable-output-
escaping="yes">&lt;String&gt;</xsl:text><xsl:text>                    </xsl:text>getAll<xsl:value-of
select="ClassName"/>Names() {
          ArrayList<xsl:text  disable-output-escaping="yes">&lt;String&gt;</xsl:text>  vAll<xsl:value-
of     select="ClassName"/>Names     =     new     ArrayList<xsl:text     disable-output-
escaping="yes">&lt;String&gt;</xsl:text>();
            //Check if database connected
          Collection col = getCollectionEXist(collectionSim);
          boolean fileExistsInDatabase = fileExistsInDatabase(col);
          if (fileExistsInDatabase) {
              //Query database for <xsl:value-of select="ClassName"/>Name
              String     query     =     "//<xsl:value-of     select="ClassName"/>s//<xsl:value-of
select="ClassName"/>//Name";
             NodeList nodeList = xpathQueryXMLDatabase(col,  query);
                int length = nodeList.getLength();
                for    (int    onNode    =    0;    onNode<xsl:text    disable-output-
escaping="yes">&lt;</xsl:text>length; onNode++) {
                  Node node = nodeList.item(onNode);
                  String str = node.getTextContent();
                  if (!vAll<xsl:value-of select="ClassName"/>Names.contains(str)) {
                      vAll<xsl:value-of select="ClassName"/>Names.add(str);
                  }
                } //End for
             } //End fileExistsInDatabase
        return vAll<xsl:value-of select="ClassName"/>Names;
    } …..
<xsl:for-each select="./OperationParameter">
   <xsl:apply-templates select = "."/>
</xsl:for-each>
  @WebMethod(operationName = "<xsl:value-of select="./ClassName"/>Name")
  public String <xsl:value-of select="./ClassName"/>Name() {
       return <xsl:value-of select="./ClassName"/>Name;
  } ……..
   </xsl:template>
</xsl:transform>
```

Figure 4-4 Code Snippet of Reference Architecture in XSLT Stylesheet

The reference architecture provides the instructions that form the input to the transformation API for XML (see section 4.3.5.3) in order to generate the Java source code.

4.3.5 Prototype of the IT Infrastructure Transformation

A transformation process is used to automatically generate a prototype of the IT infrastructure based on a textual analysis of the application domain ontology (see section 4.3.5.1), creating a database schema with instance data (see section 4.3.5.2) and transformation process using the transformation API for XML (see section 4.3.5.3).

4.3.5.1 Prototype Data Structure

A textual analysis of the application domain ontology is carried out to identify the nouns and verbs that are mapped to the IT infrastructure components such as class, operation and parameter. The context of this implementation chapter is the Service Oriented Architecture (SOA). As part of SOA, the classes, operations and parameters are mapped to Web Services, Web Methods and Web Parameters respectively.

Initially a lexical analysis is performed on each RDF statement in the application domain ontology. Lexemes are created from the semantic meaning of the subject, predicate and object of each RDF statement. A Lexeme is a sequence of characters in the source program that represent an instance of a token (Aho, Lam, Sethi, & Ullman, 2006). The semantic meaning of the subject and object is derived from the class of the RDF individual of that subject and object. The resultant lexemes represent the Web Service, Web Method and Web Parameter tokens. A token is a key word that the Prototype of the IT Infrastructure Transformation Component processes. Taking the RDF statement *JohnSmith teaches SoftwareDevelopment* for example. *JohnSmith* has the semantic meaning of *Lecturer*. *Lecturer* represents an identifier for the name of the Web Service IT infrastructure component. *Teaching* represents an identifier for the Web Method within the Web Service. *SoftwareDevelopment* has the semantic meaning of *Module*. *Module* represents as an identifier for the Web Parameter of the Web Method. The lexemes are represented in the Prototype Data Structure XML document as illustrated in Figure 4-5.

```
<Class>
    <ClassName>Lecturer</ClassName>
    <OperationParameter>
      <Operation>Teaches</Operation>
      <Parameter>Module</Parameter>
    </OperationParameter>
  </Class>
```

Figure 4-5 Prototype Data Structure

The Prototype Data Structure XML document is created using JDOM ("JDOM v1.1," 2007). JDOM provides a Java-based solution for accessing, manipulating, and outputting XML data. The Prototype Data Structure XML document is stored in the eXist database located on port 8680.

The lexemes that represent the tokens in the XML document are plugged into template code from the reference architecture in order to generate source code that can be compiled and executed using the transformation process described in section 4.3.5.3.

4.3.5.2 Component Repository

The Component Repository contains the database schema and instance data derived from the stakeholder's requirements.

A database schema is generated from the application domain ontology based on XML Schema ("W3C XML Schema v2.0," 2004). An XML document based on the database schema is generated and populated with instance data from the application domain ontology as illustrated in Figure 4-6. Interaction with the database involves the technologies XQuery ("XQuery 1.0: An XML Query Language (Second Edition)," 2010), XUpdate ("XUpdate update," n.d.) and XPath ("XML Path Language (XPath) Version 1.0," 1999). XQuery is a query language that uses the structure of XML to express queries stored in XML. XUpdate is a lightweight XML query language for modifying XML data. XPath, the XML Path Language, is a query language for selecting nodes from an XML document.

```
<Lecturers>
   <Lecturer>
      <Name>JohnSmith</Name>
      <teaches>
         <Module>SoftwareDevelopment</Module>
      </teaches>
   </Lecturer>
   <Lecturer>
      <Name>JohnRyan</Name>
      <teaches>
         <Module>TeamProject</Module>
      </teaches>
   </Lecturer>
</Lecturers>
```

Figure 4-6 XML Schema and Instance Data

The Web Services from the prototype of the IT infrastructure can access their instance data from XML documents in the Component Repository stored in the eXist database on port 8680.

4.3.5.3 Transformation API for XML

The Transformation API for XML (TrAX) ("Using the JAXP Transform APIs," n.d.) provides Java applications with the ability to transform XML documents based on transformation instructions described in XSLT stylesheets. The goal is to transform the XML document into Java source code specifically Web Services. The reference architecture (cf. section 4.3.4) contains the instructions for carrying out the transformation and the template source code. The Prototype Data Structure XML document is based on the lexemes that represent the tokens from the application domain ontology (cf. section 4.3.5.1). The XSLT processor in TrAX transforms the rototype Data Structure XML document and XSLT instructions into Java source code.

The Java source code is stored in a skeleton NetBeans project ("NetBeans IDE v6.8," n.d.). Apache ANT ("Apache Ant v1.8.2," 2010) is used to compile, build and deploy the NetBeans project to a glassfish server on port 8081. The deployment of the Java source code in conjunction with the component repository (cf. section 4.3.5.2) represents the prototype of the IT infrastructure.

The stakeholders can test the Web Services as presented in Figure 4-7. Figure 4-8 shows the SOAP request and SOAP response after information was provided through the Web Service Tester. Stakeholders and also view the instance data stored in XML documents on the noSQL database eXist as shown in Figure 4-9.
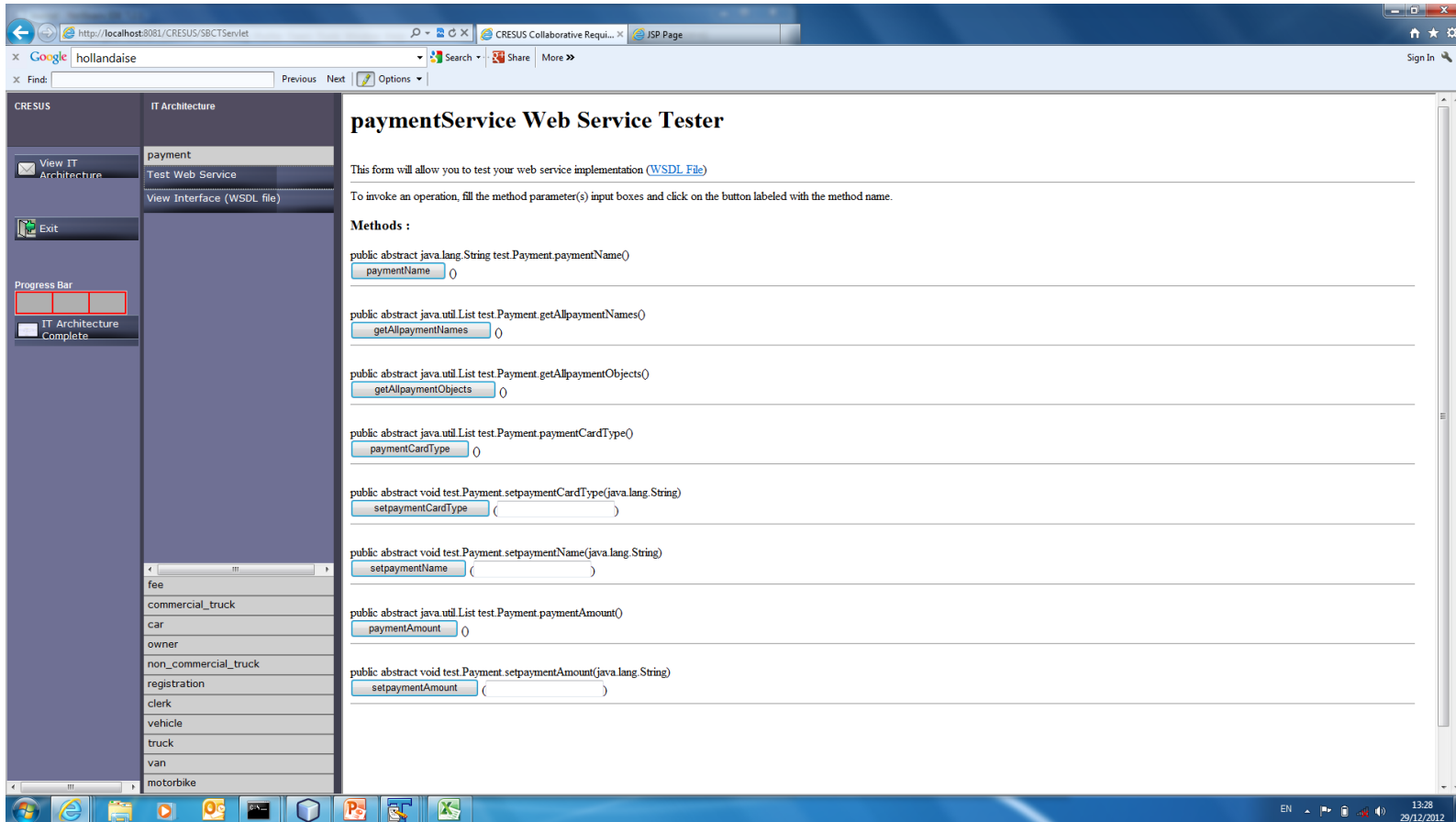
Figure 4-7 Automatically Generated Web Service

Figure 4-8 SOAP Request and Response
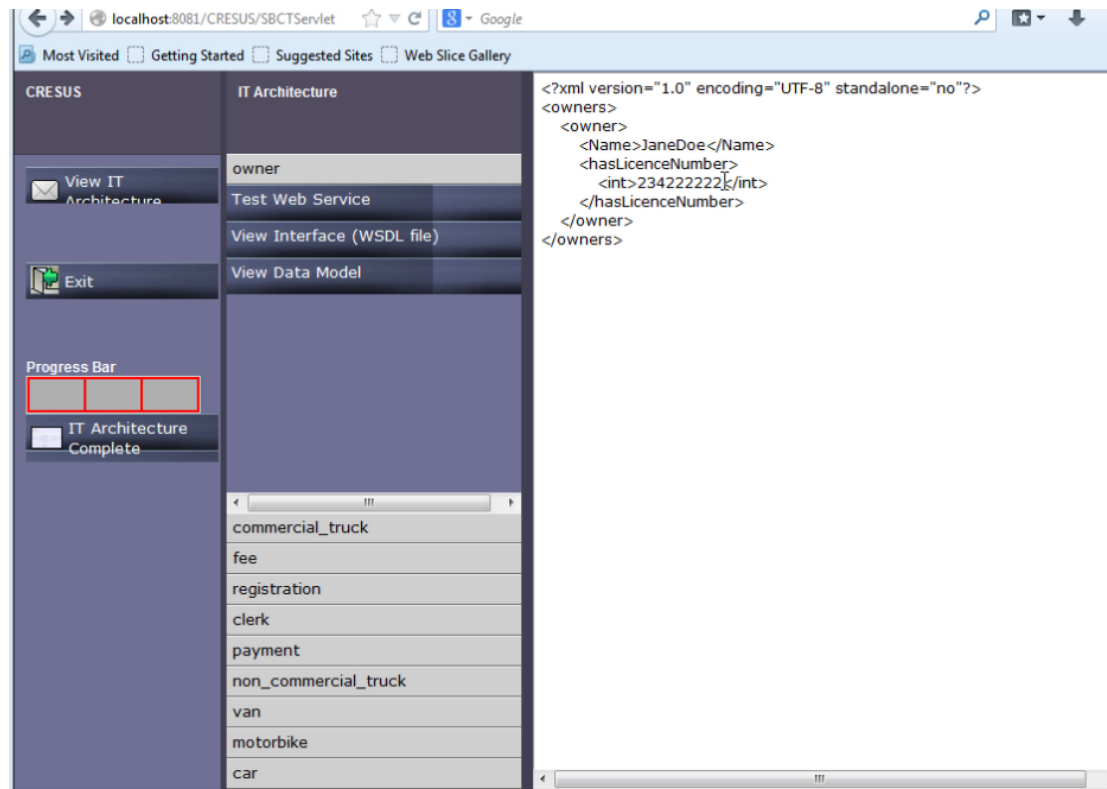
Figure 4-9 XML File Stored in NoSQL Database (eXist)

4.3.6 Model Alignment

The aim of theModel Alignment component is to align the IT infrastructure components to the business process model.

A Java algorithm is used to map the class and operations identified in the prototype data structure (cf. section 4.3.5.1) to the BPMN ontology (cf. section 4.3.3).

An XPath query on the XML document that represents the prototype data structure identifies the class names of the Web Services. In BPMN a pool is the graphical representation of a participant in a collaboration (OMG, 2011 p. 112). Participants can be realised by a Web Service (OMG, 2011 p. 116). Therfore the class name is mapped to both the pool and participant in the BPMN ontology.

The model alignment component automatically generates the RDF statement as described in Figure 4-10. Taking *Lecturer* as an example of a web service the model alignment component generates the following RDF individuals and statements:-

- Roll_Out_of_the_IT_Software_Image is an RDF individual of class business_process_diagram.
- Lecturer is an RDF individual of class pool.
- Lecturer is an individual of class participant_name.
- RDF statement "Roll_Out_of_the_IT_Software_Image has_business_process_diagram_pools Lecturer" is generated.
- RDF statement "Lecturer has_pool_process_ref Lecturer_Process" is generated.

business_process_diagram(b1)	
pool(pi)	i = 1..Number of class names
participant_name (pi)	i = 1..Number of class names
has_business_process_diagram_pools (b1, pi)	i = 1..Number of class names
has_pool_process_ref(pi, concat (pi, "_Process"))	i = 1..Number of class names

Figure 4-10 BPMN Assertions

The operation names are identifiable from an XPath query on the XML document that represents the prototype data structure. The operation names

represent Web Methods that a Web Service provides. A service task in BPMN is a task that uses a service such as a Web Service (OMG, 2011 p. 158). The service task can invoke operations that send and receive messages from Web Services through WSDL operations (Christensen, Curbera, Meredith, & Weerawarana, 2001). The Web Method are exposed as WSDL operations. Therefore the Web Methods of a Web Service are mapped to BPMN service tasks of a BPMN Participant. The model alignment component automatically generates the RDF statement as described in Figure 4-11.

service_task(concat("getAll", pi, "Names"))	i = 1..Number of class names
service_task(concat("getAll", pi, "Objects"))	i = 1..Number of class names
service_task(concat(pi, "Name"))	i = 1..Number of class names
service_task(concat("set", pi, "Name"))	i = 1..Number of class names

Figure 4-11 BPMN Service Task Assertions

Taking *Lecturer* Web Services with the *teaches* WebMethod as an example, the model alignment component generates the following RDF individuals and statements:-

- getAllLecturerNames is an RDF individual of class service_task
- getAllLecturerObjects is an RDF individual of class service_task
- LecturerName is an RDF individual of class service_task
- setLecturerName is an RDF individual of class service_task
- teaches is an RDF individual of class service_task

The BPMN ontology is rendered by the Diagram Render component (see section 4.3.7) and displayed in a graph format. The stakeholders can create additional BPMN statements through the controlled natural language interface (cf. section 4.3.1).

4.3.7 Diagram Renderer

The Diagram Render component renders a business process in diagrammatic form similar to Figure 4-12.
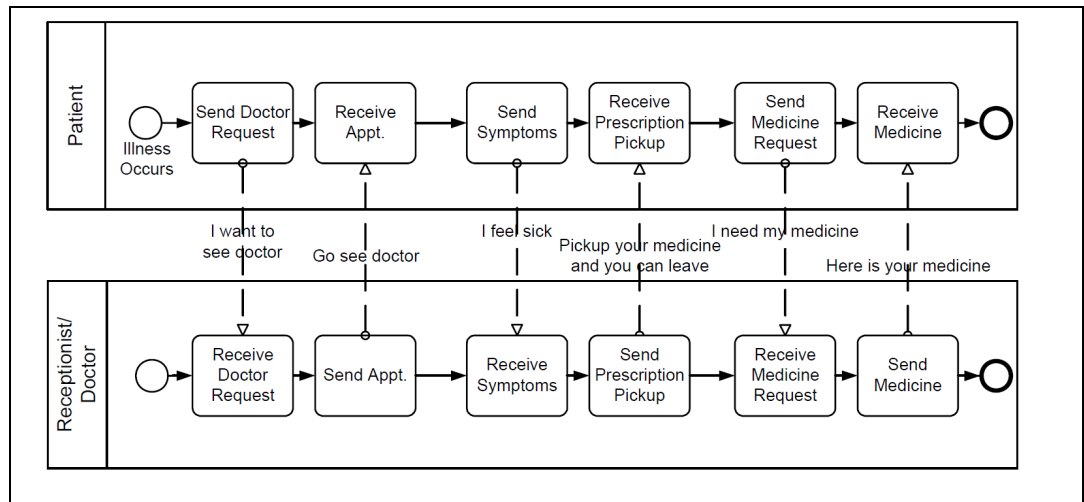
Figure 4-12 Business Process Diagram (OMG, 2011)

The rendering of the business process diagram is based on mapping the elements in the BPMN Ontology to Vertices or Edges in a graph as described in Table 4-1. The graph is created using mxGraph ("mxGraph v1.10.0.2," 2012). mxGraph is a leading open source Java graph visualisation and layout component. The implementation of mxGraph is on the server side of CRESUS-T. mxGraph is a 3rd party product used in the CRESUS-T architecture.

Table 4-1 Mapping BPMN Ontology Elements to Graph Components

BPMN Ontology Element	Graph Component
Pool/Swimlane	Vertice
Task	Vertice
Process	Vertice
Event	Vertice
Gateway	Vertice
Data Object	Vertice
Annotation	Vertice
Group	Vertice
Message Flow	Edge
Sequence Flow	Edge

The Modelling and Analysis activity of the CRESUS process model that involves modelling a business process in CRESUS-T is shown in Figure 4-13. Figure 4-14 shows a magnified business process diagram.

Figure 4-15 demonstrates the creation of a message flow within CRESUS-T and

Figure 4-16 shows the controlled natural language interface that guides the stakeholder through the creation of a BPMN message flow.

Figure 4-13 Modelling a Business Process in CRESUS-T

Figure 4-14 Business Process Diagram in CRESUS-T
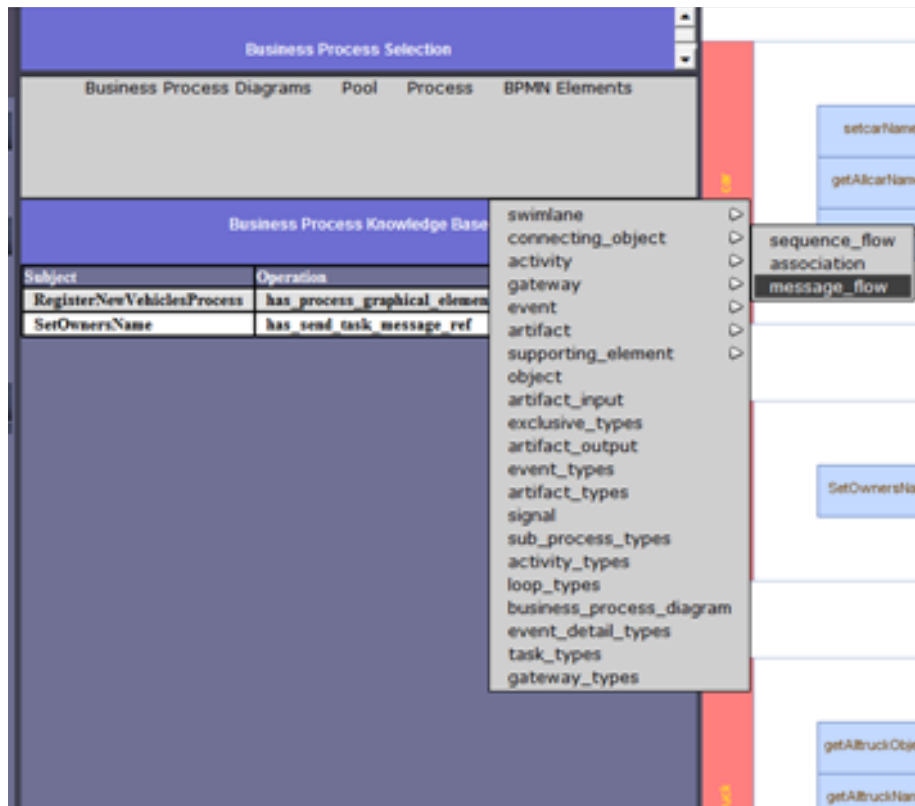
Figure 4-15 Creating a Message Flow in CRESUS-T

Figure 4-16 Controlled Natural Language Interface in CRESUS-T
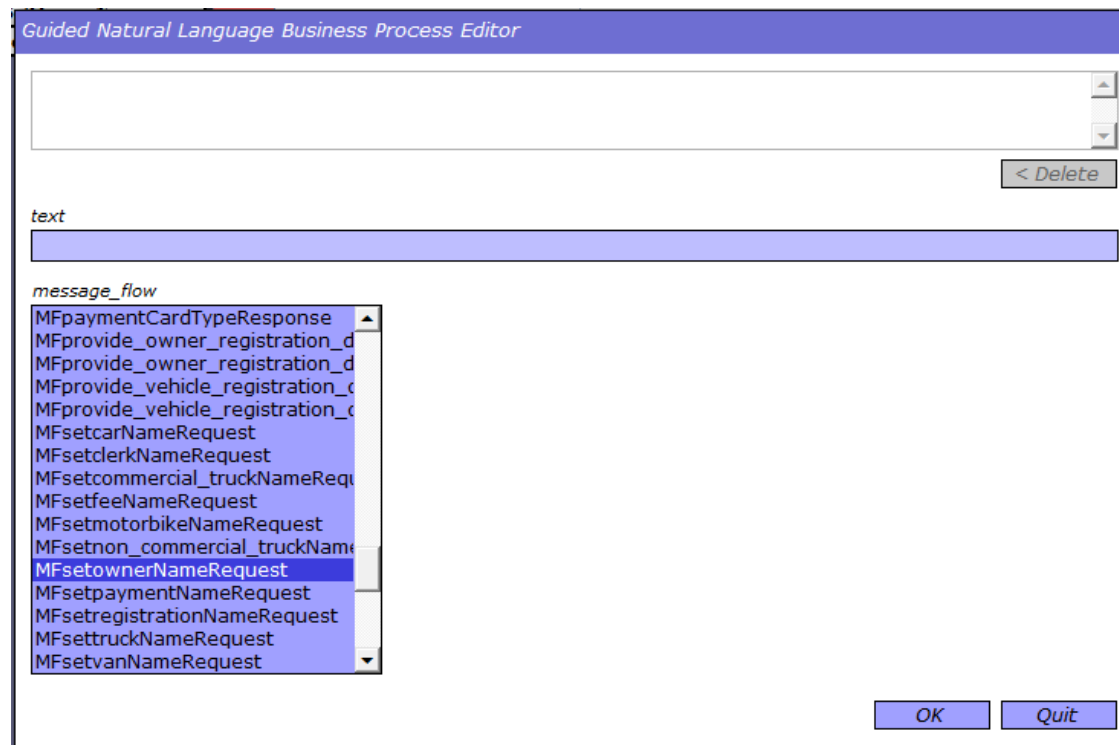
4.4 Summary

This chapter discussed the implementation of the collaborative requirements elicitation framework called CRESUS that consists of the CRESUS process model and support tool CRESUS-T.

The technical architecture of the support tool, CRESUS-T was presented. The technical approach around controlled natural communication, automatic creation of a prototype of the IT infrastructure, alignment of the BPMN model with IT infrastructure and rendering of the BPMN diagram were described in detail.

The CRESUS framework was implemented to support a real-world case study of a business process for the roll out of the IT software image at the National College of Ireland, in order to provide a platform for evaluating the framework in chapter 5.

5 Evaluation of the CRESUS Framework

5.1 Introduction

This chapter details the methodology and analysis of the results from three experimental studies in order to evaluate the extent to which support for shared understanding amongst stakeholders is fostered through the proposed features of the CRESUS framework. The studies were applied to a real-world situation in NCI. The real-world setting chosen was the rollout of the IT software image business process within NCI. This business process is representative of a typical business process in any organisation that has issues around collaboration and communication resulting in delays with completing the process.

The motivation for the proposal of the CRESUS framework was determined by examining the need to support business executives' communication and modelling in the alignment of the business process and IT infrastructure. The first experimental study involved a scenario-based prototype that provided a user interface, and allowed business executives to specify their organisational needs in natural language as they moved through a pre-planned sequence of actions and mouse clicks. Section 5.2.1 presents details of this study.

The CRESUS Framework comprising of the CRESUS process model and the new support tool CRESUS-T, were evaluated in experimental studies two and three. The CRESUS process model provides four activities to guide the stakeholders during requirements elicitation (cf. section 3.2).

Experimental study two comprised of the three activities of the CRESUS process model namely Domain Understanding, Domain Modelling and Requirements Elicitation. In the context of global software development, it is important for the stakeholders to initially meet face to face in order to encourage team formation which helps to strengthen collaboration and communication. As such the first and second activity of the CRESUS process model, Domain Understanding and Domain Modelling were evaluated face to face. The third activity of the CRESUS process model, Requirements Elicitation, was implemented in the support tool, CRESUS-T (cf. section 3.2.2), and evaluated asynchronously. Section 5.2.2 provides details of experimental study two.

Experimental study three comprised of the two central activities represented by the Requirements Elicitation, and Modelling and Analysis activities of the CRESUS process

model. In order to complete the Modelling and Analysis activity, there is a need to align the IT infrastructure with the business process. The creation of the IT infrastructure is an integral part of the Requirements Elicitation activity. Therefore, experimental study three repeated the evaluation of the Requirement Elicitation activity of experimental study two, in addition to the Modelling and Analysis activity. Experimental study three is described in section 5.2.3.

5.2 Methodology

The features of the CRESUS framework namely support for collaboration, communication, prototyping and modelling, were evaluated in the experimental studies as described in Table 5-1.

Table 5-1 Mapping of CRESUS Framework Features to Experimental Studies

Features	Experimental Study		
	1	2	3
Collaboration		X	X
Communication	X	X	X
Prototyping		X	X
Modelling	X		X

Table 5-2 provides an overview of the experimental studies. The methodology follows the classic notation system provided by Campbell and Stanley (1963, p. 6) where

- X represents an exposure of a group to an experimental variable, the effects of which are to be measured.
- O represents an observation or measurement recorded on an instrument.
- R represents the random assignment of employees to experimental and control groups.

Table 5-2 Overview of Experimental Studies

	Experimental Study 1	Experimental Study 2	Experimental Study 3
Aim	To examine the desire of business executives for a framework that supports communication and modelling.	To examine to what extent the CRESUS framework both aided communication in the development of a shared understanding and supported collaborative requirements elicitation to bring about organisational, and associated IT infrastructural changes.	To examine to what extent the CRESUS framework supported collaborative requirements elicitation through enhancing shared understanding of potential organisational and associated IT infrastructure requirements around asynchronous communication and business process modelling.
Rational	To investigate if there is a real need in industry for a collaborative requirements elicitation framework that employs an ontology-driven natural language communication mechanism. This mechanism may be used to gather requirements based on organisation's needs, where these needs are translated into a simulation of a business process supported by the IT infrastructure as represented by web services.	To investigate if the CRESUS framework increases communication through an ontology-driven natural language communication mechanism that captures an organisation's needs. In addition, to determine if the automatic generation of a prototype that represents the IT infrastructure supports the organisational needs faithfully. All of which leads to collaborative decision making.	To investigate if employees collaborate, and communicate asynchronously with other employees in their team to develop a shared understanding of the IT infrastructure that would support the business process of the application domain.

CRESUS Process Model Activities		1. Domain Understanding 2. Domain Modelling 3. Requirements Elicitation	3. Requirements Elicitation 4. Modelling and Analysis
Subjects	Ten senior and middle managers from institutes of higher education in Ireland were invited to take part in the experimental study. Four were from NCI, three from Institutes of Technology (IOT) Blanchardstown, one from IOT Carlow, one from IOT Tallaght and one from IOT Waterford.	Thirteen employees from NCI were invited to take part in the experimental study. Eight employees were from the School of Computing and five were from the IT Department.	Eighteen employees from NCI were invited to take part in the experimental study. Sixteen employees were from the School of Computing and two were from the IT Department.
Methodology	Pre-experimental design (One Shot Case Study) was used as a research methodology Business Executive Group X———————O The senior and middle managers were in the roles of vice president (one participant), head of school (seven subjects) and head of department (two subjects).	Randomised matched pairs design was used. One employee that was the process owner was interviewed and twelve employees were assigned to four groups. Group A R——O——O——X——O——O Group B R——O——O————O Group C R——O——O——X——O——O Group D R——O——O————O	Randomised matched pairs design was used. Employees were assigned to two groups. Group A R——O——X——O——X——O——O Group B R——O——X——O————O——O Each group consisted of nine employees in the role of business executive, IT architect and business users. All groups were administered a pre-test, an assessment of their BPMN knowledge, post-test and an interview. Both groups received BPMN

		Each group consisted of three employees in the role of business executive, IT architect and business user. All groups were administered both a pre-test and post-test. Training on the support tool CRESUS-T was provided to the experimental groups A and C only.	online training. Training on the support tool CRESUS-T was provided to the experimental group A only.
Instrumentation	Questionnaire	Interview, workshop, online questionnaires, logs, observations, and a debriefing session.	Online questionnaires, journals, logs, observations, interviews, and a debriefing session.
Treatment	Scenario-based Prototype	Experimental Groups A, and C were provided with CRESUS-T and email. Control Groups B, and D were provided with email only.	Experimental Group A was provided with CRESUS-T and email. Control Group B was provided with email and Google Docs.
Duration of experiment	Two weeks to complete the questionnaire	Interview [30 minutes] Pretest [5 minutes] Discussion on organisation change scenario [10 minutes] Workshop [30 minutes] Controlled Experiment [1 hour] Post test [5 minutes] Total [2 hours and 20 minutes]	Pretest and Posttest Questionnaires [30 minutes] BPMN tutorial and test [1 hour] CRESUS-T tutorial [40 minutes] Controlled experiment at least 10 minutes over one week [at least 50 minutes] Interview [30 minutes] Debriefing [30 minutes]

			Total at least 4 hours over two weeks

5.2.1 Experimental Study 1

The CRESUS framework for collaborative requirements elicitation that combines collaboration, communication, prototyping and modelling was specified in section 3.2. However, the state of the art was unclear if such a framework would be of interest to senior and middle managers with the ability to influence change in an organisation such as institutes of higher education. The outcome of this experimental study would define the future research direction for the development of the CRESUS framework. Therefore, it was necessary to evaluate a mock-up of a scenario-based prototype to determine the business executives' desire for such a framework.

The first experimental study evaluated the desire of business executives for the communication and modelling features introduced in the CRESUS framework that aims to support requirements elicitation. This section provides an overview of the study.

5.2.1.1 Overview

There are twenty six funded higher education institutes in Ireland (Higher Education Authority, 2014). Cycyota & Harrison (2006) found that the mean response rate for organisational researchers surveying executives is declining, with an overall rate of 32%. Given the small number of HEA funded institutes and the low response rates with executives, it was decided to adopt a convenience sample as part of the sampling strategy. Convenience sampling is based on sampling people who are easy to survey (Mitchell & Jolley, 2004). A weakness with convenience sampling is that the subjects may not represent a significant portion of the population (Mitchell & Jolley, 2004). To mitigate the weakness, the subjects were purposefully selected based on critical case sampling. Critical case sampling is defined as the process of selecting a small number of important cases that are likely to yield the most information and have the greatest impact on the development of knowledge (Patton, 2001 p. 236). Morse (2000) suggests four factors that will result in fewer subjects being needed to reach data saturation. The following four factors were taken into consideration to determine the sample size:

- The focused nature of the study (Opinion of senior and middle managers at institutes of higher education).

- The clear nature of the topic area (the need to support business executives' communication and modelling in the alignment of the business process and IT infrastructure).
- The high quality of data (Opinion of senior and middle management).
- The use of shadowed data. (Fink (1998) is an example of obtaining the opinion of owners and managers on factors representing modern IT adoption. The response rate in the survey was 33% which is also similar to Cycyota & Harrison (2006) results of 32%).

Cycyota & Harrison (2006) suggest alternatives to increase the response rate of executives such as contact through existing social networks of executives. The population names for the convenience sample were identified by colleagues at the National College of Ireland and the University of Dublin, Trinity College. Contact details of the population where identified from the internet and approached directly through email followed by a phone call. This study used a single stage sampling procedure. A single stage sampling procedure is one in which the researcher has access to names in the population and can sample the people directly (Creswell 2008, p. 148).

The experimental study used a one shot case study, pre-experimental design (Creswell, 2008 p. 158). A pre-experimental design involves studying a single group with an intervention during the experimental study followed by a measure. This does not have a control group to compare with the experimental group. The pre-experimental design involved a group of managers that were exposed to the scenario-based prototype followed by the Interface Systems Usability Questionnaire (see Appendix 1).

The procedure used for experimental study one is presented in Table 5-3.

Table 5-3 Procedure for Experimental Study One

Step	Task
1	Identify subjects
2	Subjects go through scenario-based prototype
3	Subjects complete Interface Systems Usability Questionnaire
4	Reminder email to complete the Interface Systems Usability Questionnaire
5	Analysis of answers to Interface Systems Usability Questionnaire

The subjects that took part in the study were ten senior and middle managers from institutes of higher education in Ireland. Four managers were from NCI, three managers

were from Institute of Technology (IOT) Blanchardstown, one manager from IOT Carlow, one manager from IOT Tallaght and one manager from IOT Waterford. The mangers were in the roles of vice president (one subject), head of school (seven subjects) and head of department (two subjects).

A scenario-based prototype was created around the specific scenario of planning changes to faculty and programmes within a higher education setting. The prototype consisted of a PowerPoint mock-up that simulates a natural language user interface whereby the business executive specifies their organisational goal and rules in English through a pre-planned sequence of actions and mouse clicks. The scenario-based prototype was used in order to ensure repeatability and to allow the subjects to use it remotely. In addition, this type of prototype was chosen as part of the experimental study for the initial stages of the research as it offers a means of getting quick and frequent feedback on a system.

The Interface Systems Usability Questionnaire (see Appendix 1) was chosen as the preferred type of data collection procedure for this study because of the need to get subjects opinion (Mitchell and Jolley, 2004 p. 180). The questionnaire comprised thirty items that were principally based on a 1-5 point semantic differential scale (LaLomia and Sidowski 1990) and open comments sections. The major content sections of the Interface Systems Usability Questionnaire are Demographics; Perceived Usefulness; Usability Heuristics; User Interface Satisfaction; Screen; Learning; Project Specific questions; List the negative aspects; List the most positive aspects; List any suggested improvements; and List any scenarios that are more relevant to you. Several items were adopted from Davis (1989) Perceived Usefulness, Davis (1989) Perceived ease of use, Nielsen (1993) Usability Heuristics, and Chin (1988) User Interface Satisfaction. The questionnaire consists of questions like "The use of a Goal editor for entering an organizational goal in natural language is desirable" and "The systems automatic interpretation of the goal in terms of business process and rules is desirable". The business executives were asked to rate their opinion on the 5 dimensions of the scale ranging from "I strongly agree" to "I strongly disagree".

The aim of the questionnaire was to examine:-

- The need for a system that allows business executives communicate business needs using natural language.

The goal of capturing this information was to determine the business executives desire to communicate requirements using natural language. This metric was important for evaluating the motivation for developing a framework based around a communication feature.

- The need for a system that can automatically interpret the business needs that relate to a business process.

The goal of capturing this information was to determine the business executives desire to automatically interpret the business needs that relate to a business process. This metric was important for evaluating the motivation for developing a framework based around a modelling feature.

The analysis was conducted around communication and modelling features of the CRESUS framework. The evaluation of communication and modelling is described in section 5.3.1.1 and section 5.3.1.2 respectively.

5.2.2 Experimental Study 2

The results of experimental study 1 influenced the design of the CRESUS framework (see section 5.3.1). The properties for the CRESUS process model were specified in section 3.2.1. A series of properties for a support tool that implemented the two central activities of the CRESUS process model were specified in section 3.2.2 and CRESUS-T was implemented.

The CRESUS-T support tool was designed to allow subjects collaborate and communicate requirements asynchronously using natural language, and automatically generate a prototype of the IT infrastructure. The state of the art showed that there is a need for:-

- Facilitating the activities of a collaborative process model through tool support (cf. section 2.3.1).
- Achieving effective communication (cf. section 2.3.2).
- Understanding of the impact of software architectural choices on the ability to satisfy current and future non-functional requirements (cf. section 2.2.2).

The aim of the second experimental study was to examine the extent to which the CRESUS framework both aided communication in the development of a shared understanding and supported collaborative requirements elicitation to bring about organisational, and associated IT infrastructural changes.

The variables, dimensions and instruments for experimental study two are presented in Table 5-4.

<div align="center">Table 5-4 Research Variables Dimensions and Instruments Study Two</div>

Variables	Dimensions	Instruments
Level of Service	Job Title	Level of services and academic qualifications pre-test questionnaire
	Job Role	Level of services and academic qualifications pre-test questionnaire
	Number of years of service with NCI	Level of services and academic qualifications pre-test questionnaire
Academic Qualifications	NFQ Level	Level of services and academic qualifications pre-test questionnaire
Collaboration	Collaborative Decision Making	Asynchronous communication for decision making post-test questionnaire
	Understanding of the IT infrastructure increased due to collaboration	Asynchronous communication for decision making post-test questionnaire
Communication	Frequency of Communication	CRESUS-T Logs
Prototyping	Prototype supports organisational change	Simulation based communication tool (CRESUS) post-Test questionnaire
Shared Understanding	Shared understanding of how IT can support the organisational changes	Simulation based communication tool (CRESUS) post-Test questionnaire

This section provides an overview of the experimental study two. An analysis of the results is described in section 5.3.2.

5.2.2.1 Overview

The School of Computing at NCI had created a conversion programme in response to Government tenders to provide courses that would up-skill the workforce. The conversion programme required desktop computers with specialised software to run the practical laboratory sessions. The IT Department are responsible for the business process that involves creating the IT software image and rolling it out to the desktop computers. The IT software image contains the specialised software required by the conversion programme. The business process had issues around collaboration and communication resulting in delays with completing the process. The IT Department wanted to develop the IT infrastructure that would automate the business process and engaged in a requirement gathering exercise. This environment provided an opportunity to evaluate the CRESUS framework for eliciting requirements of the future system that would align to the business process. The setting for this experimental study is described in section 4.2.

The experimental procedure used for experimental study two is presented in Table 5-5.

Table 5-5 Procedure for Experimental Study Two

Step	Task
1	Identify subjects.
2	***Domain Understanding***
2.1	Interview NCI Senior IT Administrator.
2.2	Analyse the results of the Interview (Problem Domain Description).
3	***Domain Modelling***
3.1	Invitation to complete pre-test level of service and qualifications questionnaire was sent to all subjects through email.
3.2	Administer pre-test level of service and qualifications online questionnaire.
3.3	Analyse results of the pre-test.
3.4	Assign subjects to matched pairs.
3.5	Randomly assign subjects of each matched pair to the experimental groups A and C, and control groups B and D.

3.6	Invitation to attend a meeting to discuss the case study of the organisational change scenario that will be used during the experimental study.
3.7	Conduct meeting that describes the case study used during the organisational change scenario.
3.8	Invitation to attend workshop with brainstorming technique was sent to all subjects through email.
3.9	Conduct workshop with brainstorming technique.
3.10	Analyse results of the Workshop (Application Domain Ontology).
4	*Requirements Elicitation*
4.1	Invitation to attend experiment sent to all subjects through email.
4.2	Expose treatments to experimental groups A and C, and control groups B and D.
4.3	Administer post-test asynchronous communication for decision making questionnaire to the experimental groups A and C, and control groups B and D.
4.4	Administer post-test simulation-based communication tool questionnaire to the experimental groups A and C.
4.5	Compare performance of the experimental and control groups on the post-test(s) using tests of statistical significance.

All employees from the School of Computing and the IT Department in NCI were contacted to identify if they would participate in the study. Thirteen employees that consisted of eight employees from the School of Computing and five from the IT department agreed to take part in the study. One employee that participated was the business process owner and only took part in the interview. The remaining twelve employees participated in the experimental study.

The aim of the Domain Understanding activity of the CRESUS process model was to understand the problem and the application domain of the future system. The design (cf. 3.2.1.1) showed that there is a need to understand the application domain by studying key documents, and interviewing stakeholders. The implementation of the CRESUS process model at NCI showed that there is no documentation on the roll out of the IT software image business process (cf. section 4.2). As such it was necessary to interview the senior IT administrator in NCI to determine the problem domain description for the

roll out of the IT software image. The interview was conducted in a natural work environment in NCI for 30 minutes. The interview comprised of nine questions that assessed the operation of the business process (see Appendix 2). A recording and transcript of the interview are included in CD-ROM that accompanies this thesis. The problem domain description artefact that represented the business process for the roll out of the IT software image was created from an understanding arising out of the interview. The problem domain description was validated with the senior IT administrator and used as input to the next Domain Modelling activity of the CRESUS process model (see section 5.3.2.2.1). An analysis of the outcome of the interview is presented in section 5.3.2.2.

Due to the small number of subjects in the experimental study, it was deemed important to review any threats to the validity of the experiment. Campbell and Stanley (1963) identified extraneous variables that threaten the internal validity of the experiment such as assignment bias, and selection. Creswell (2008, p. 163-165) outlines actions that can be taken to minimise the threats.

Assignment bias may occur if all employees that are IT administrators in the IT Department were in the same group. Likewise, if all employees that are course directors in the School of Computing were in the same group, and if all employees that are lecturers in the School of Computing were in the same group. In collaborative requirements elicitation the stakeholders are typically comprised of business executives, IT architects and business users of the system. In the context of this experimental study, there was a mapping between business executives and course director, IT architect and IT administrators, business users of the system and lecturers. The action that was taken in response to the assignment bias threat was to use a block design. A block design (Bailey, 2008 p. 51) is used to remove the effects of a few of the most important nuisance variables that may cause variability in the experimental study.

A second type of threat to the internal validity of the experiment related to selection. Subjects can be selected who have certain characteristics that predispose them to have certain outcomes. The action to take in response to this threat is to assign the matched subjects randomly to the groups. Randomization is then used to reduce the contaminating effects of the remaining nuisance variables. This approach with a block design and randomisation leads to greater accuracy in the experiment and boosts the power of the experiment with fewer subjects.

The experimental design for this study used a randomised matched pairs design. A matched pairs design is a special case of a block design. As part of the matched pairs design, twelve of the employees were exposed to the level of service and qualifications questionnaire pre-test.

The aim of the level of service and qualifications questionnaire was to evaluate the blocking factors based on factual information such as the employee's role in NCI, level of service with NCI and academic qualifications based on the National Framework of Qualifications (NFQ) of Ireland (Quality and Qualifications Ireland, 2012). The role allowed the creation of homogenous blocks that determined if the employees were course directors, IT administrators or lecturers. The level of service allowed the creation of homogenous blocks that determined the employees years of service within NCI ranging from 1-2 years, 3-4 years, 5-6 years, 7-8 years, 9-10 years and greater than 10 years. The academic qualifications allowed the creation of homogenous blocks that determined the employees NFQ level at level 8 (Honours Degree), level 9 (Master's Degree) and level 10 (PhD Degree). The control information that related to employee's role, level of service, and academic qualifications were used in the analysis for matching employees was derived from the pre-test questionnaire's data.

Eight employees from the School of Computing and four employees from the IT Department were assigned to the matched pairs on the basis of their scores in the pre-test questionnaire. The employees were randomly assigned based on one subject of each pair to the experimental group A and C, and the other subject to the control group B and D (see Table 5-2). An analysis of the results from the level of service and qualifications online questionnaire is described in section 5.3.2.1.1.

The employees were invited to a meeting to discuss the organisational change scenario and the roles of each team member. A case study detailing organisational change scenario, objective of the study, and the employees role during the experiment was provided to the employees based on their roles of business executive, IT architect and business user (referred to as staff in the scenario) (see Appendix 2). The task for the employees was to collaborate with their group to identify and agree on the instance data based on the concepts defined in the Domain Modelling activity of the CRESUS process model. They had to identify and create a suitable IT system for storing the data. During this study, group members were only allowed to communicate asynchronously.

The aim of the Domain Modelling activity of the CRESUS process model was to create a formal naming and definition of the types, properties, and interrelationships of the concepts from the application domain. The state of the art showed that there is a need for an ontology-driven process model that drives the remaining activities of the process model (cf. section 2.2.1). The design showed that this could be accomplished using a workshop that uses the *brainstorming technique* (cf. section 3.2.1.2). As such it was necessary to evaluate the Domain Modelling activity of the CRESUS process model with the employees in a workshop to define the ontology. In order to define the ontology, the problem domain description (see section 5.3.2.1) was examined to determine the concepts, and actions as represented by relationships, that formed part of the application domain ontology. The employees were invited to take part in a workshop (see Appendix 2). The workshop was conducted in a natural work environment in NCI over a 30 minute period. The design (cf. section 3.2.1.2) showed that the workshop employing the *brainstorming technique* is useful for rapidly generating as many concepts and actions in a short period of time. As the ontology for each group is similar, only an analysis of the outcome from group A's workshop is described in section 5.3.2.3.

The aim of the Requirements Elicitation activity of the CRESUS process model was to gather the requirements and generate and test a prototype of the IT infrastructure. The subjects were invited to attend the experiment in the ICELT research laboratory. However due to a logistical reason the participants completed the experimental study in a natural work environment in their own offices at NCI. The experiment took place over a one hour period. A treatment was administered to the experimental and control groups as outlined in the methodology section of Table 5-2. The experimental groups A and C were administered the treatment of the support tool, CRESUS-T and email. The control groups B and D used email only. A log of all communication between the subjects was stored in a communal email for each group. In addition, the tool support, CRESUS-T logged all communication messages. The information that related to the communication feature used in the analysis was derived from the communal emails and CRESUS-T's logged data. CRESUS-T logs and email responses between employees were evaluated to identify:

- The frequency of communication.

The goal of capturing this information was to determine if the CRESUS framework leads to increased communication. This metric was important as frequent communication helps create a shared understanding between stakeholders regarding the importance of technology in supporting the business needs.

The asynchronous communication for decision making online questionnaire (see Appendix 2) was administered to experimental groups A and C, and control groups B and D. The post-test contains thirteen items in four major content sections. The content sections are demographics, frequency of communication, quality of communication and perception. The majority of these questions asked for factual information such as the person's name, positive aspects of the experiment, negative aspects of the experiment, and suggested improvements to the experiment. The question on subject's perception is based on a one to five semantic differential scale from "strongly disagree" to "strongly agree". The information that related to the collaboration, and prototyping features were derived from the subjects' perception of collaborative decision making, and a shared understanding of the IT infrastructure supporting organisational change, used in the analysis is derived from this questionnaire's data. The online questionnaire was chosen as a data collection procedure for the study because of the rapid turnaround in data collection and the proficiency of subjects with technology. The major content sections evaluated the collaboration feature to identify:-

- If CRESUS framework helps the employees make collaborative decisions.
- If employees understanding of the IT infrastructure changes to support the organisational goals has increased due to collaboration.

The goal of capturing this information was to determine if the CRESUS framework supports collaborative decision making and if the subjects understanding of the IT infrastructure changes has increased due to collaboration. These metrics were important as collaboration is a predictor of shared understanding. It demonstrates if the CRESUS framework supports collaborative work practices that subjects can successfully execute for themselves.

The experimental groups A and C were then administered the simulation based communication tool post-test questionnaire (see Appendix 2). The questionnaire contains thirty six items. Items were taken from Davis (1989) Perceived Usefulness, Davis (1989) Perceived ease of use, Nielsen (1993) Usability Heuristics, and Chin (1988) User

Interface Satisfaction. The items that did not provide meaningful information for this scenario were discarded. The remaining questions were modified to a five point semantic differential scale. The major content sections of this survey are demographic details, perceived usefulness, usability heuristics, user interface satisfaction, screen layout, learning and project specific questions. The major content sections of this survey around perceived usefulness evaluated:

- If the prototype supports organisational change faithfully.

The goal for capturing this information was to determine if the automatic generation of a prototype that represents the IT infrastructure represents a realistic solution. This metric was important as it demonstrates an ability to plug real IT systems such as web services into the prototype which can lead to richer requirements and is the basis for aligning the IT infrastructure with the business process as shown by the state of the art (cf. section 2.2.3).

- If the CRESUS framework helped the employees create a shared understanding of how IT infrastructure can support the organisational changes.

The goal for capturing this information was to determine if the automatic generation of a prototype that represents the IT infrastructure supports the business process during organisational change. This metric was important as shared understanding may be demonstrated by an artefact such as the IT infrastructure.

The major content sections of the simulation based communication tool post-test questionnaire also evaluated shared understanding to identify:-

- If the employees create a shared understanding of how IT can support the organisational changes

The goal for capturing this information was to determine if the employees perceive that the framework fosters shared understanding of how the IT infrastructure can support the organisational change. This metric was important as it provides qualitative information that reinforces the support for shared understanding.

The performance of the experimental and control groups were compared in the post-test(s) using tests of statistical significance in terms of collaborative decision making, frequency of communication, prototype supports organisational change, and shared

understanding. IBM's SPSS tool ("IBM - SPSS software - Ireland," 2015) was used to manage quantitative and qualitative data. An analysis of the results of CRESUS-T logs, email and the post-test are described in sections 5.3.2.4.

5.2.3 Experimental Study 3

Experimental study two (cf. section 5.3.2) evaluated the collaboration, communication and prototyping features of the CRESUS process model. Experimental study three repeated the evaluation of the Requirement Elicitation activity of experimental study two in addition to the Modelling and Analysis activity. The properties for the Modelling and Analysis activity were specified in section 3.2, and further described in section 3.2.1.4. The properties for the design of the support tool, CRESUS-T that facilitated the Modelling and Analysis activities were specified in section 3.2.2 and further described in section 3.2.2.3. The state of the art showed that there is a need to create enterprise models from web services that forms the basis of aligning the business process to the IT infrastructure (cf. section 2.2.3). Therefore it is necessary to conduct an explicit evaluation of the subjects experience with collaboration, communication, prototyping, and modelling features of the CRESUS framework to determine if the objectives of this research have been fulfilled.

The modelling feature of the CRESUS framework may be assessed by scrutinising model recall, problem understanding and subjects experience with modelling. In addition, the collaboration feature of the CRESUS framework may be assessed by scrutinising the subjects' experience with collaboration. The communication feature of the CRESUS framework may be assessed by scrutinising the frequency of communication and the subjects experience with asynchronous communication. In addition, the prototyping feature of the CRESUS framework may be assessed by scrutinising subjects' opinion that the prototype of the IT infrastructure has the potential to represent an authentic solution. Shared understanding may be scrutinised the subjects experience with the approach of reaching a shared understanding.

The aim of the third experimental study was to investigate the extent to which the CRESUS framework supported collaborative requirements elicitation through enhancing shared understanding of potential organisational and associated IT infrastructure requirements around asynchronous communication and business process modelling.

The variables, dimensions and instruments for experimental study three are presented in Table 5-6.

Table 5-6 Research Variables Dimensions and Instruments for Study Three

Variables	Dimensions	Instruments
Level of Service	Job Title	CRESUS pre-test questionnaire
	Job Role	CRESUS pre-test questionnaire
	Number of years of service with NCI	CRESUS pre-test questionnaire
Academic Qualifications	NFQ Level	CRESUS pre-test questionnaire
Domain Knowledge	Knowledge of the Rollout of the IT software image	CRESUS pre-test questionnaire
Modelling Knowledge	BPMN knowledge	CRESUS pre-test questionnaire
	Other modelling knowledge	BPMN assessment
Collaboration	Employee's experience about collaboration during the creation of the data model, IT infrastructure and business process model	Interview
Communication	Frequency of Communication	CRESUS-T logs
	Employee's experience of asynchronous communication around completing the data model, IT infrastructure and business process model	Interview
Prototyping	Employee's opinion that the IT infrastructure has the potential to represent an authentic solution	Interview
Modelling	Employee's experience of Modelling a Business Process	Interview
	Model Recall	CRESUS post-test questionnaire
	Problem Solving	CRESUS post-test questionnaire
Shared Understanding	Employee experience with the approach of reaching a shared understanding during the creation of the data, IT	Interview

	infrastructure and business process model	

This section provides an overview of the experimental study. An analysis of the results is described in section 5.3.3.

5.2.3.1 Overview

The setting for experimental study three is the same as experimental study two (see section 5.2.2.1). This experimental study was conducted in a natural work environment in NCI over the period of one working week where employees participated for at least 10 minutes every.

The experimental procedure used for experimental study three is presented in Table 5-7.

Table 5-7 Procedure for Experimental Study Three

Step	Task
1	Identify subjects.
2	Administer CRESUS pre-test online questionnaire.
3	Analyse results of the pre-test.
4	Assign subjects to matched pairs.
5	Randomly assign subjects of each matched pair to experimental group A and control group B.
6	Provide online tutorial on BPMN to all subjects.
7	Administer BPMN assessment to all subjects.
8	Analyse results of the BPMN assessment.
9	Provide a series of online tutorials on CRESUS-T.
10	Provide case study one to control group B and case study two to experimental group A.
11	Administer treatments to experimental group A and control group B.
12	Administer CRESUS post-test online questionnaire to all subjects.
13	Administer interview to all subjects.
14	Compare performance of the experimental and control groups on the post-test and interview using tests of statistical significance.

Eighteen employees that consisted of sixteen employees from the School of Computing and two employees from the IT Department took part in the experimental study. The employees were given an information sheet for participants (see Appendix 3) that outlined the background of the research and procedure of the study. The subjects completed an informed consent form (see Appendix 3) agreeing to take part in the study.

The employees were administered the CRESUS pre-test online questionnaire that was designed to remove the effects of a few of the most important nuisance variables such as the role of the employee, level of service, and academic education, in addition to domain knowledge, and modelling knowledge. The blocking factor for domain knowledge allowed the creation of homogenous blocks that determined if employee's level of knowledge with the application domain was novice or expert based on a five point Likert scale from "I have no knowledge" to "I have expert knowledge". The blocking factor for modelling knowledge allowed the creation of homogenous blocks that determined the subjects' confidence at using modelling techniques such as business process model and notation based on five point Likert scale from "not confident" to "very confident". The blocking factors that related to employee's role in the organisation, level of service with NCI, educational qualifications, domain knowledge, and modelling knowledge were used in the analysis for matching subjects was derived from the pre-test questionnaire's data. The employees were assigned to matched pairs on the basis of their scores in the measures that related to role, number of years in employment with NCI, academic qualifications based on the NFQ, domain knowledge and modelling knowledge (see Appendix 3). One employee of each matched pair was randomly assigned to the experimental group A, and the other employee to the control group B. An analysis of the results from the level of service and qualifications online questionnaire is described in section 5.3.3.1.1.

An online tutorial on Business Process Modelling Notation (BPMN) was provided. A motivation for this approach was that employees may be geographically located in different places during the experimental study and this approach would ensure consistency of training. The online BPMN tutorial was created using screen recording and video editing software, Camptasia Studio ("Camtasia, Screen Recorder and Video Editor v8," n.d.). The tutorial was created based on a PowerPoint presentation (see Appendix 3) and a script (see Appendix 3). The tutorial described in detail the fundamental concepts underlying BPMN and the techniques for applying them. The

recording was nine minutes in duration and was uploaded to YouTube (*BPMN Introduction*, 2013).

The employees were administered BPMN assessment that examined their knowledge of BPMN (see Appendix 3). The BPMN assessment contained eight questions in two major content sections. The content sections were level of confidence and modelling understanding. The test required employees to respond to a series of questions that intended to measure their confidence at using business process model and notation based on 5 point likert scale from "not confident" to "very confident". Other questions intend to measure the participant's knowledge of BPMN. The knowledge questions contained a BPMN symbol and several selection buttons with labels that represent different BPMN symbols such as "Event", "Activity", and "Gateway". An analysis of the results from the BPMN assessment is described in section 5.3.3.1.2.

A series of online tutorials on how to use the CRESUS-T support tool were provided to the employees assigned to experimental group A. The online tutorial for the CRESUS-T support tool were created using screen recording and video editing software, Camptasia Studio ("Camtasia, Screen Recorder and Video Editor v8," n.d.). The tutorials were created based on a script (see Appendix 3) and series of PowerPoint presentations. The recordings were uploaded to YouTube (CRESUS Approach Part 1, 2013; CRESUS-T Data Entry Part 2, 2013; CRESUS Approach IT Infrastructure Part 3, 2013; CRESUS Approach BPMN Model Part 4, 2013). The CRESUS-T tutorials, script, and PowerPoint presentations are included in CD-ROM that accompanies this thesis.

The control group was given case study one that described the business process for the roll out of the IT software image, details on the conversion programme, explanation of the employee roles, the assignment of employees to the roles, and where to locate the online journal (see Appendix 3). The employees had to describe on a daily basis, two highlights and two challenges in the online journal. Experimental group A were given case study two. Case study two was the same as case study one, and contained additional information such as details of where to locate CRESUS-T and tutorials on how to use CREUSUS-T. The employee's tasks were to communicate asynchronously with other employees in their team to develop a shared understanding of the IT infrastructure that would support the business process of the application domain. This included identifying any data stored in the IT infrastructure. This part of the experimental study took place over a one week period in a natural setting and in conjunction with their other work

responsibilities. They were requested to work on the experiment every day over the working week for at least 10 minutes between the hours of 10:00 and 12:00.

The experimental group A was administered the treatment of the CRESUS-T support tool, and email. The control group B were provided with email and Google Docs. A log of all communication between the employees was stored in a communal email for each group. In addition, the CRESUS-T logged all communication messages. The information that related to the communication feature used in the analysis was derived from the communal emails and CRESUS-T's logged data. CRESUS-T logs, email responses between employees and interactions on Google Docs were evaluated to identify:

- The frequency of communication.

The goal for capturing this information was to determine if the framework leads to increased communication. This metric was important as frequent communication helps create a shared understanding between technology providers and business users regarding the importance of technology in supporting the business activities. The analysis of the results is presented in section 5.3.3.4.1.

The experimental group A and control group B was administered the CRESUS post-test online questionnaire (see Appendix 3). The aim of the post-test online questionnaire was to examine model recall and problem solving. For model recall it is premised that the higher the model recalls, the higher the level of understanding of the application domain being modelled. The major content section of this questionnaire around model recall examined:-

- The employees' ability to recreate the business process model including the interaction with any IT systems and identify any data used during the experiment.

Problem solving required employees to use the mental understanding they developed during the experiment, to suggest answers to a similar problem. The major content section of this questionnaire around model understanding examined:-

- The employees' ability to create a model of a business process model that is different from the case study.

The goal for capturing this information was to determine if the employees have developed a deeper understanding of the requirements. This metric is important because problem solving question require the employees to use the mental understanding that they developed during the experiment which would suggest more sophisticated cognitive understanding of the application domain and the higher is the level of understanding they have developed.

All employees were administered a recorded interview (see Appendix 3) of the employees experience or opinion of communication, business process modelling, shared understanding and collaborative requirements elicitation. The interviews were categorized as a full positive experience, full negative experience or a mixture of positive and negative experience.

The major content section of the interview contained questions that examined:-

- If employees experience with collaboration during the creation of the data model, IT infrastructure and business process model was a positive experience when CRESUS-T and email were used compared to when email and Google Docs were used.

The goal for capturing this information was to determine if the employees have a positive experience with collaboration. This metric was important because it demonstrates that collaboration needs to be designed into a process model.

- If employees experience with asynchronous communication was positive.

The goal for capturing this information was to determine if the employees have a positive experience with asynchronous communication. This metric was important in the context of global software development where it is impractical to meet face to face some of the time and other forms of asynchronous communication become predominant such as email.

- If the prototype of the IT infrastructure has a high fidelity to commercial or open source software.

The goal for capturing this information was to determine if the IT infrastructure model was realistic. This metric was important as it demonstrates an ability to allow requirements engineers specify non-functional requirements for applications that invoke

at least some real web services. This provides the employees with a realistic method for analysing the architectural choice.

- If employees experience with collaborative business process modelling was positive.

The goal for capturing this information was to determine if the employees have a positive experience with modelling. This metric was important as it indicates if the framework supports modelling.

- If experience with the approach of reaching a shared understanding of the data model, IT infrastructure and business process model was positive.

The goal for capturing this information was to determine if the employees perceive that the framework fosters shared understanding. This metric was important as it provides qualitative information that reinforces the support for shared understanding.

The performance of the experimental and control groups on the post-test(s) were compared using tests of statistical significance in terms of collaboration, communication, prototyping, modelling and shared understanding.

QSR-NVivo ("Qualitative Research | Data Analysis Software | NVivo," n.d.) was used to manage and analyse the qualitative data from various data collection techniques such as interviews, emails, journals and observations. The data was cleaned, coded and arranged into specific themes that generally cantered around negative and positive experiences or opinions. IBM's SPSS tool ("IBM - SPSS software - Ireland," 2015) was used to manage quantitative data. The results of the analysis are described in section 5.3.3.

5.3   Results and Analysis

This section presents in detail an analysis of the results obtained from three experimental studies.

The results from experimental study one are based on an evaluation of a scenario-based prototype by mangers from institutes of higher education in Ireland. The analysis of results is presented in section 5.3.1.

The results from experimental study two are based on an interview, workshop and evaluation of CRESUS framework by employees using pre-tests, CRESUS-T logs, and post-test(s) instruments. An analysis of the results is presented in section 5.3.2.

The results from experimental study three are based on an evaluation of the CRESUS framework by employees of NCI using a pre-test, an assessment of BPMN, CRESUS-T logs, a post-test and an interview. An analysis of the results is presented in 5.3.3.

5.3.1 Experimental Study 1 Results and Analysis

The data gathered during experimental study one was analysed in terms of the communication and modelling features that should be supported by the CRESUS framework.

5.3.1.1 Communication Feature

The majority of senior and middle managers (70%) have confirmed that there is a need for a solution that allows goal definition through natural language is desirable. The majority of senior and middle managers (80%) have confirmed that there is a simple and natural dialogue when using the prototype. The manager's comments about the natural language interaction provided by the scenario-based prototype confirm the desire for such a support tool. A sample of comments from the business executives were: "The idea seems to be very good, especially the natural language interface, which I particularly like" and "Use of English to set goals". The business executives indicated that the use of "Natural Language" was a positive aspect of the study.

5.3.1.2 Modelling Feature

The majority of senior and middle managers (70%) have confirmed that the system's automatic interpretation of the goal in terms of business process and rules is desirable. One business executive commented that a positive aspect of the scenario-based prototype was that the "System will interpret the goal and identify the organisational process for appropriate service"

5.3.1.3 Discussion and Findings

While this was only a limited survey the results are promising. The business executives intuitively understood the approach being proposed:-

1. To communicate business goals in natural language.
2. To make use of a support tool that automatically interprets those goals.
3. To identify a business process that can deliver the appropriate services.

Moreover, they strongly indicated that they believed such an approach was desirable. The first experimental study concluded that the prototyped communication and modelling features are desirable by business executives in a requirements elicitation framework.

5.3.2 Experimental Study 2 Results and Analysis

The methodology for experimental study two is described in section 5.2.2. The methodology described the procedure in this study with details on the evaluation instruments namely, the level of service and qualifications online questionnaire, interview, workshop, CRESUS-T logs, asynchronous communication for decision making online questionnaire and the simulation based communication tool online questionnaire (see Appendix 2).

The level of service and qualifications online questionnaire was examined in order to determine the matched pair and randomisation of employees to experimental groups A and C, and control groups B and D. An analysis of the results is presented in section 5.3.2.1.

The NCI senior IT administrator was interviewed in order to determine the business process for the roll out of the IT software image. The outcome of the interview is a problem domain description and is presented in section 5.3.2.2.

The NCI employees participated in a workshop in order to formally describe the application domain of the future system. The outcome of the workshop was an ontology and is presented in section 5.3.2.3.

The NCI employees participated in an evaluation of the CRESUS framework in order to determine if the CRESUS framework increases communication through an ontology-driven natural language mechanism that captures an organisation's needs. In addition, to determining if the automatic generation of a prototype that represents the IT infrastructure supports the organisational needs faithfully. All of which leads to collaborative decision making. The evaluation of the CRESUS framework used the

CRESUS-T logs, email, asynchronous communication online questionnaire and the simulation based communication tool online questionnaire. An analysis of the results is presented in section 5.3.2.4.

5.3.2.1 Prologue

This section analyses the results of experimental study two in terms of the level of service and qualifications questionnaire. The level of service and qualifications questionnaire contains three control variables *role*, *level of service*, and educational qualifications. Role represented the employee's job role in NCI. Level of service indicates the degree of years that the employee is in employment in NCI. Educational qualification was operationalized by encompassing one dimension *NFQ Level*. NFQ level indicates the degree to which the employee are educated to such as at level 8 (Honours Degree), level 9 (Master's Degree) and level 10 (PhD Degree).

5.3.2.1.1 Matched Subjects

Table 5-8 describes the employee's role, title, number of years of service with the National College of Ireland and their level of educational qualification attained based on the National Qualification Framework. Academic's 2, 3, 5 and 8 with a title of lecturer grade II and where their role indicated that they are programme directors were assigned to the business executive role. The academics 7, 11, 10 and 12 whose titles were lecturer grade I, lecturer grade II, postdoctoral research fellow and support tutor, and where their role indicates that they lecture in the School of Computing were assigned to the business user role. The IT personnel 1, 4, 6 and 9 with the titles senior IT administrator, IT support specialist and IT support personnel were assigned to the IT architect role. The employees were pair-matched based on their number of years of service with the NCI and their level of educational qualification attained based on the national qualification framework. The matched employees were 2 and 3; 7 and 11; 4 and 6; 5 and 8; 10 and 12; and, 1 and 9. Employees were randomly allocated to the experimental and control groups as follows,

- Employees 2, 7 and 1 to experimental group A.
- Employees 3, 11 and 9 to control group B.
- Employees 8, 10 and 4 to experimental group C.
- Employees 5, 12 and 6 to control group D.

Discussion and Findings

The match between experimental group A and control group B is excellent with one difference in educational qualifications of the employees in the role of the IT architect.

The match between experimental group C and control group D indicates that the cumulative experience gained in work benefits the control group D, whereas the cumulative experience in educational levels benefit the experimental group C. On balance years of experience may make up the difference with the educational level.

Table 5-8 Matching Employees Experimental Study 1

Role	Name	Title	Number of years of service with NCI						The level of education attained based on the National Qualification Framework			Group
			1-2 years	3-4 years	5-6 years	7-8 years	9-10 years	>10 years	Honours Degree	Masters	PhD	
Programme Director	2	Lecturer II				X				X		A
	3	Lecturer II				X				X		B
	5	Lecturer II			X						X	D
	8	Lecturer II					X			X		C
Lecture in School of Computing	7	Lecturer I	X						X			A
	10	Computing Support Tutor	X							X		C
	11	Post Doctoral Research Fellow	X						X			B
	12	Lecturer II						X		X		D
IT Personnel	1	Senior IT Administrator				X				X		A
	4	IT Support		X					X			C
	6	IT Support		X					X			D
	9	IT Support Specialist				X			X			B

5.3.2.2 Domain Understanding

This section analyses the results of experimental study two in terms of the Domain Understanding activity of the CRESUS process model.

5.3.2.2.1 Problem Domain Description

The following problem domain description is an outcome of the interview with the IT Senior Administrator.

The IT Department of the National College of Ireland (NCI) undergo a business process for rebuilding a software image on an annual basis. The software image is then put on all desktop computers in the computer laboratories in NCI. The software image contains the software that is required for each module on all programmes offered by NCI.

An IT administrator is responsible for initiating the business process when they send an email to all lecturers in each school around March or April. The email contains the previous year's software requirement list. The software requirement list contains details about the software application and its version number. Lecturers in each school have four to six weeks where they can review the software that they require for the next academic year.

The Lecturers will check each module's current software requirements. They will then review the module software requirements for the next academic year. The lecturer's module software requirements are compared to the software requirement list. If there are new software requirements or changes to existing versions of software then the lecturer can update the module software requirements and the software requirement list with the software that is required for their module for the next academic year. For open source software, the lecturer has to also download the software to a global directory. The updated software requirement list is returned to the IT administrator.

However the software requirement list never comes back on the due date. IT sends a reminder email at least three times and sometimes it is escalated to the Dean of School.

The IT administrator starts to build the software image around the end of May, beginning of June. To build a Citrix server farm from the ground up could take six weeks. This involves taking one server out of a live environment. The server is wiped of software, performing a raid and mirroring, followed by an operating system, Microsoft patches,

Citrix patches, and then the IT software image based on the software identified from the software requirements list. In addition, different policies are applied and scripts are executed. There may be some compatibility issues for example PhPDev cannot run on Citrix. The issue is around running a server on a server. Tests are carried out on the build to ensure that all the software works and that the software is compatible. If the software doesn't work then it has to be tested in isolation. The computer laboratories are upgraded over the years and as a result there will be different hardware platforms that have to be tested with the software. For example one lab may contain Dell GX520 and 270's in another lab. The image is then rolled out to all the desktop computers in the computer laboratory.

The email that IT sends around March or April emphasises that this is the one and only one time that a request for software can be made. Installing software outside of the IT software image doesn't work well. The problem that may arise is when a virus is detected on a desktop computer in the laboratory. That computer laboratory could be quickly re-imaged. However software outside of the IT software image is lost. Occasionally exceptions do occur where a lecturer may require a piece of software for ten or fifteen students in their class. The software can be installed in the first two rows of a laboratory, under the provision that should something happen in the laboratory, and the computers are re-imaged, then the software will be lost. The installation of additional software is a man-intensive activity, where several people form the IT department will go down and install the software on each machine. The students that will then use that application have to be set up with the necessary privileges to run the software.

5.3.2.3 Domain Modelling

This section analyses the results of experimental study two in terms of the Domain Modelling activity of the CRESUS process model.

5.3.2.3.1 Application Domain Ontology

The following description is an outcome of the workshop with experimental group A. This outcome is representative of the outcomes form experimental group C and control groups B and D. As such only the outcome of experimental group A's workshop is discussed in this section.

The objective of the Domain Modelling activity was to create an ontology of the application domain. The concepts and relationships identified by the employees in experimental group A were modified due to limitations with the technology used in CRESUS-T as follows:-

1. Dots changed to dash e.g. JRE1.6 becomes JRE1-6 (limitation with ACE).

2. Cannot start with numbers must start with a String (limitation with ACE).

3. Cannot enter special characters e.g. &, $ and so on (limitation with ACE).

4. Strings cannot be entered with spaces between them (limitation with Protégé).

5. Properties must be unique such as the property "requires" was specified in several places and for uniqueness had to be changed to requiresSome, requiresAn, requiresMore (limitation with CRESUS-T).

6. Limitation with experiment: There cannot be two triples with the same subject and object and different predicate. For example, the following statements have the same subject and object but different predicates and therefore there is a need to delete one of the statements:
Dreamweaver requiresMore RAM => Software requiresMore Hardware
Explorer canNotbeUsedOnA Mac => Software canNotbeUsedOnA Hardware
(limitation with CRESUS-T).

7. The Concept "Resource" is a keyword and cannot be used (limitation with CRESUS-T).

During the workshop the experimental group A identified concepts, relationships, rules and instance data from the problem domain description. The output of the workshop was modified based on the limitations with the software. The follow represents the modified output of the workshop taking into account the limitations with the software, where the limitations are highlighted by underlining or deleting changes :-

Concepts

– Lecturers, Classrooms, Computers, Whiteboard, Interactivity, Software, Lab, Hardware, Modules, SupportStaff, Resources, Students.

– SoftwareVersion, OperatingSystem, HardwareConstraints, AccessRights.

Relationships (triples)

– Lecturer ~~Academic~~ teaches Module.

– Modules allocatedTo Classrooms.

– Module requires Hardware.

– Modules requiresSome Software.

– Module requiresAn OperatingSystem.

– Software hasVersion SoftwareVersion.

– Computer has HardwareConstraints.

– Software ~~requires~~ grants AccessRights.

– Software requiresMore Hardware.

- Software isProhibitedFor Students.
- Lecturer ~~Academic~~ teaches Module.
- Lab contains<u>ANumberOf</u> Computers.
- HardwareLab contains Hardware.
- Computers host Software.

Rules (constraints)

- If a lab ~~hosts~~ containsANumberOf Computers, and Computers host Software -> Lab ~~has~~ with Software.
- Lab contains X number of Computers.

Instance data

- ComputerArchitecture requires Breadboard.
- Paul<u>Stynes</u> teaches WebDevelopment.
- WebDevelopment requiresANumberOf Computer.
- WebDevelopment requiresSome MobileAPI.
- WebDevelopment allocatedTo Room2-01 ~~2.01.~~
- Dreamweaver requiresMore RAM.
- ~~Explorer canNotbeUsedOnA Mac.~~
- IE7 is prohibited for <u>All</u>Students.
- Java has JRE1-6 ~~1.6.~~
- PhPDev ~~requires~~ grants AdministrationRights.

Who approves the requirements as part of the collaboration?

- Business User enters requirements.
- Business Executive approves the requirements.
- IT Architect approves the requirements.

5.3.2.4 Requirements Elicitation

This section analyses the results of experimental study two in terms of the Requirements Elicitation activity of the CRESUS process model.

5.3.2.4.1 Collaborative Decision Making and IT Infrastructure Collaboration

Collaborative Decision Making indicates the degree to which *employees perceive that decisions were made in a collaborative way* from strongly agree to strongly disagree based on asynchronous communication for decision making online questionnaire. IT infrastructure collaboration indicates the degree to which *employees perceive that their understanding of the IT infrastructure increases due to collaboration* from strongly

agree to strongly disagree based on asynchronous communication for decision making online questionnaire.

100% of the experimental group that used CRESUS-T indicated that the support tool helped the group make collaborative decisions.

In Figure 5-1, 83% of the employees in the experimental groups, A and C that used CRESUS-T indicated that decisions were made in a collaborative way and 17% were neutral. 50% of the employees in the control groups, B and D indicated that decisions were made in a collaborative way and 50% were neutral. This result indicate that the employees had a 33% higher response that decisions were made in a collaborative way when CRESUS framework was used than when email was used.
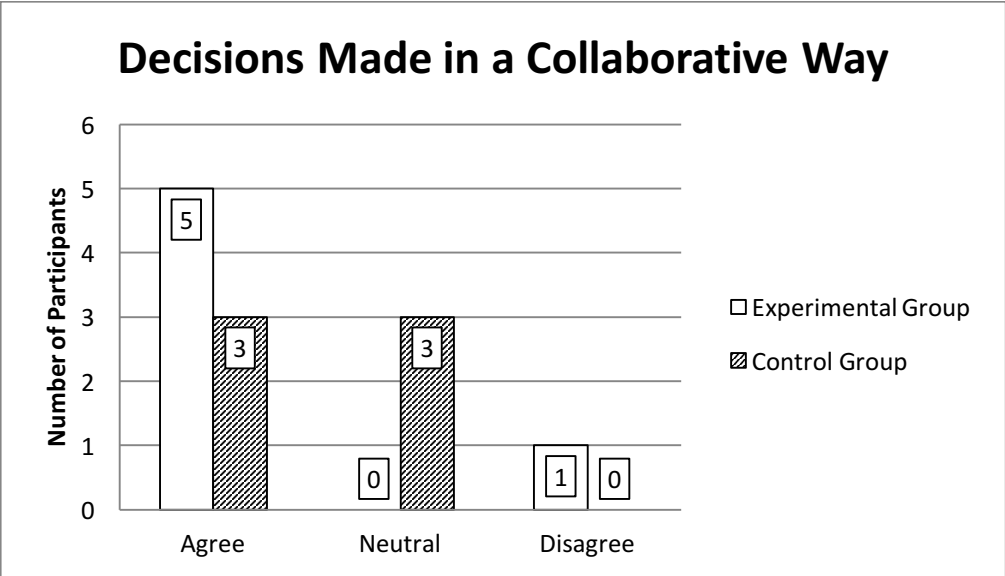
Figure 5-1 Decisions Made in a Collaborative Way

In Figure 5-2, 50% of the employees in the experimental groups A and B, indicated that their understanding of the IT infrastructure changes to support the organisational goals had increased due to collaboration. No employee in the control group indicated that their understanding of the IT infrastructure changes to support the organisational goals had increased due to collaboration, although 44% of the employees in the control groups C and D were neutral. This result indicates that the employees had a 50% higher response that their understanding of the IT infrastructure changes to support the organisational goals had increased due to collaboration when CRESUS framework was used than when other communication support tools were used.
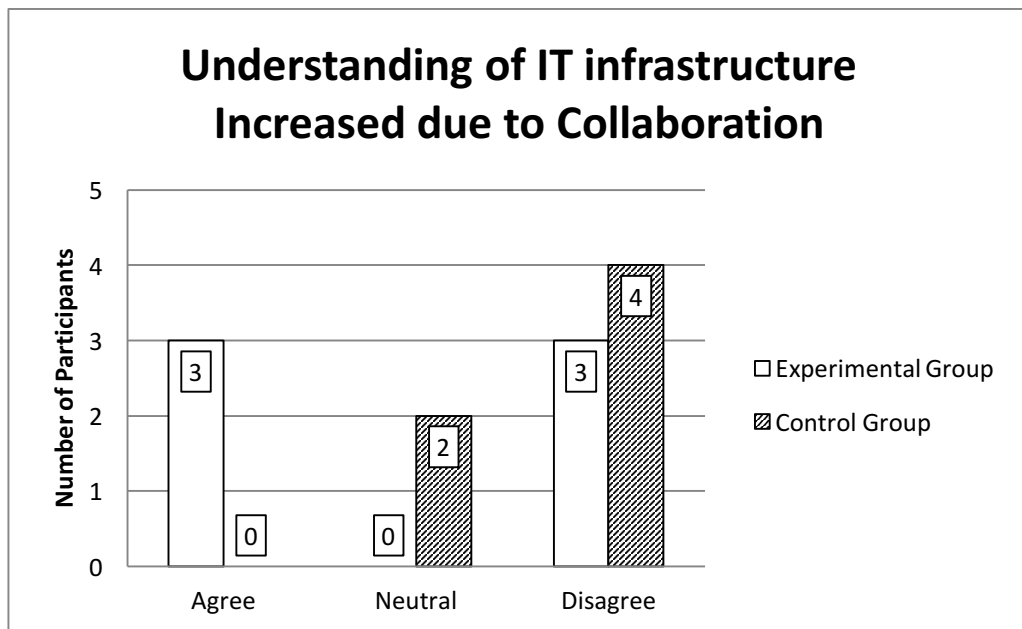
Figure 5-2 Collaboartion leads to Understanding of IT Infrastructure

The employees were asked for their opinion on collaboration. Qualitative feedback from employees that have used CRESUS-T indicates that the support tool was useful for defining requirements and facilitated collaboration. Samples of comments provided include:-

- "Good to collaborate with Informatics lecturers with regard to provision of services and facilities"
- "It enhanced collaboration and allowed the user to view rules and evaluate them 'in real time'. It is relatively easy to use once teething problems are overcome."
- "good for storing knowledge and facilitating collaboration"
- "Very useful tool for defining requirements and collaboration"

In contrast to a sample comment from the control group where they commented that they had issues with collaboration "trying to keep focus in the group".

An observation of control group B and validated by the group was that they could not made a decision on which solution to go with. This was reinforced by control group B with one employee commenting "The collaboration focused too much on the technology rather that the business".

132

5.3.2.4.2 Discussion and Findings

While this was a limited study, the results demonstrate that decisions were made in a collaborative way by employees that used the CRESUS framework. The engineering of collaborative activities such as identifying the requirements, approving the requirements, creating the IT infrastructure enabled the employees to work together to make decisions that resulted in progress beyond the control groups. This was evident by control group B getting tied up in a decision making process around the IT infrastructure at the expense of identifying the requirements.

The comment from the employee in the control group "The collaboration focused too much on the technology rather that the business" highlights the advantage with the CRESUS framework in that employees are able to concentrate on the requirements model that supports the organisational changes rather than focusing on the technology which is automatically generated.

In addition, the automatic generation of a prototype that represents the IT infrastructure provides more time for the employees to play around with the interfaces and databases of the IT infrastructure to get a better understanding of the requirements. This is validated by the analysis in that the employees had a 50% higher response that their understanding of the IT infrastructure changes in support of the organisational goals had increased due to collaboration when CRESUS framework was used.

The key finding that arises from the evaluation of collaboration Employees indicated that decisions were made in a collaborative way which was 33% higher than employees that did not use the CRESUS framework. The employees also indicated that their understanding of the IT infrastructure changes to support the organisational goals had increased due to collaboration, and this was 50% higher than the control group.

5.3.2.4.3 Frequency of Communication

Frequency of communication indicates the degree of messages that employees sent through CRESUS-T and email.

The mean frequency of communication is statistically significantly ($p <= 0.05$) higher when CRESUS-T and email were used ($\mu=25$, $\sigma= 10.218$, N=6) compared to the case when only email is used. ($\mu=20.83$, $\sigma=9.042$, N=6), paired t (5) = 2.792, p=0.038. The

effect size was large ($\rho = 0.829$). This result seems to show that the CRESUS framework encourages increased communication among employees during Requirements Elicitation activity of the CRESUS process model.

66% of the experimental group that used CRESUS-T indicated that communication that involves identifying solutions to organisational changes increases and 44% were neutral.

The employees were asked for their opinion on communication. Qualitative feedback from employees that used the CRESUS framework indicated that the support tool, CRESUS-T was useful for communicating especially around the IT infrastructure. Samples of comments provided include:-

- "Provided a new means for communication around IT infrastructure".
- "A simple communication protocol for organisational change".

5.3.2.4.4 Discussion and Findings

The state of the art showed that asynchronous communication becomes the predominant form of communication in global software development. The results presented demonstrate that eemployee's experienced significant increased asynchronous communication (p=0.038) during the Requirement Elicitation activity of the CRESUS process model. This is because the design of the CRESUS framework ensures that employees focus the communication on identifying requirements from the application domain and are supported by automatic machine translation in the generation of the IT infrastructure.

5.3.2.4.5 Prototype Supports Organisational Change

Prototype Supports Organisation Change indicates the degree to which employees that use CRESUS-T *perceive that the prototype of the IT infrastructure comprising of web services supports the organisational change faithfully* [13] from strongly agree to strongly disagree based on simulation based communication tool online questionnaire.

66% of the experimental group that used CRESUS-T indicated that the prototype of the IT infrastructure comprising of web services supports the organisational change

---

[13] where faithful (marked by fidelity to an original) "a close translation"; "a faithful copy of the portrait"; "a faithful rendering of the observed facts"

faithfully and 44% were neutral. This result was reinforce by comments from the employees such as

- "Simulation of the IT infrastructure was realistic and would support the organisational changes".
- "Good for modelling real world scenarios which illustrates to the other users (departments) a more complete workflow"

These comments are in contrast to comments from the control group. Based on observations of control group B and validated by the group, they spent a substantial amount of work in defining the IT infrastructure but very little time on the requirements of the business. They had two suggested solutions around different databases, but had not made a decision on which solution to go with. One of the employees of control group B made the following comment "Maybe the contributed [employees] should be told not to focus on specific technologies too much".

An observation of control group D and validated by the group was that they identified the software requirements list that was used in one part of the business process for the roll out of the IT software image. This list was used in a previous year. This also included the IT system that the data was stored in namely, Microsoft Excel.

5.3.2.4.6 Discussion and Findings

The comment "Maybe the contributed [employees] should be told not to focus on specific technologies too much", highlights an advantage of using CRESUS-T in that it allows the employees to focus on communication around the business and not on defining the IT infrastructure.

One of the objectives of the experimental study was to generate the IT infrastructure. One could argue that this is unrealistic and not fair considering that CRESUS-T automatically generates the IT infrastructure as part of the prototype. Further studies require employee's usage of the asynchronous support tools over a longer period of time such as one week. This would give a more realistic time for the control group to come up with a proper IT infrastructure.

With regard to the observations of control group D, they did have a simple IT system however they did have an employee that worked on the business process for the roll out

of the IT software image, in his role with the IT department. This presents a challenge with research of this nature that involves a realistic case study. Having an employee from the IT department in each group was an attempt to counter balance this situation. Overall, the results indicate that the CRESUS Framework helps create a realistic IT infrastructure that supports organisational change.

The key finding that arises from the evaluation of the employee's perception is that 66% of the employees that used CRESUS-T indicated that the prototype of the IT infrastructure comprising of web services supported the organisational change faithfully. The experimental groups felt that the simulation of the IT infrastructure was realistic and would support the organisational changes. This result is as expected because the CRESUS framework incorporates a reference architecture that is embedded in the terms from the application domain, and deploys real web services and database on a glassfish server thus simulating an authentic representation of the IT infrastructure.

5.3.2.4.7 Shared Understanding

Shared Understanding indicates the degree to which employees perceive that the *support tool helped them create a shared understanding of how IT can support the organisational changes* from strongly agree to strongly disagree based on asynchronous communication for decision making online questionnaire.

66% of the employees that used CRESUS-T indicated that the support tool helped them create a shared understanding of how IT can support the organisational changes and 44% were neutral.

5.3.2.4.8 Discussion and Findings

During the course of the experiment, the employees that used the CRESUS framework identified and approved the requirements model that represented the organisational change to automate the business process. IT represented the automatic generation of the IT infrastructure in terms of web services and data models, which was deployed to the glassfish server. This approach of creating a shared understanding through the CRESUS framework was validated by 66% of the employees.

In addition, shared understanding is manifested in the creation of the artefacts during the experiment. Experimental groups A and C created a requirements model and

automatically generated a prototype of the IT infrastructure as they went through the Requirements Gathering, and Generation and Testing sub activities of the CRESUS process. In comparison, although the control group B identified several possible IT infrastructure models, they did not have a shared understanding of the final solution due to incomplete collaborative decision making. In addition, control group B spent all their time on creating the IT infrastructure at the expense of creating a requirements model. Control group D, they reused the previous IT system used in one aspect of the business process for the roll out of the IT infrastructure. In addition, control group D created a requirements model. However their progress was aided by the prior experience of the IT architect on their team, who had experience with the business process for rolling out the IT software image. On balance, the employees that used the CRESUS framework demonstrated a higher shared understanding as manifested in the creation of the requirements model and IT infrastructure model in response to the organisational change.

5.3.3 Experimental Study 3 Results and Analysis

The methodology for experimental study three is described in section 5.2.3. The methodology describes the procedure used in this study with details on the evaluation instruments namely, CRESUS pre-test online questionnaire, online BPMN assessment, CRESUS-T logs, CRESUS post-test online questionnaire and interviews.

The CRESUS pre-test online questionnaire was examined in order to determine the matched pair and randomisation of employees to experimental group A, and control group B. An analysis of the results is presented in section 5.3.3.1.

The online BPMN assessment was examined in order to determine if the BPMN tutorial prepared the employees for modelling a business process during the experiment. An analysis of the results is presented in section 5.3.3.1.

The NCI employees participated in an evaluation of the central two activities of the CRESUS process model, Requirements Elicitation, and Modelling and Analysis. The CRESUS framework was evaluated using the CRESUS logs and CRESUS post-test online questionnaire. The employees were interviewed in relation to their experience during the experimental study. The evolution of themes that emerged from the journal entries and interviews are described in Appendix 4. An analysis of the results from the CRESUS logs, CRESUS post-test online questionnaire, journal and interview is

presented in section 5.3.3.2, section 5.3.3.4, section 5.3.3.5, section 5.3.3.7 and section 5.3.3.8.

5.3.3.1 Prologue

This section analyses the results of experimental study three in terms of the CRESUS pre-test and BPMN assessment.

The CRESUS pre-test is operationalized by encompassing the dimensions presented in the level of service and qualifications online questionnaire (see section 5.3.2.1) and two additional dimensions namely ***Domain Knowledge*** and ***Modelling Knowledge***. Domain knowledge indicates the degree to which employees have expert knowledge or no knowledge of the application domain. Modelling knowledge indicates the degree to which employees are confident at modelling or not confident at modelling.

The BPMN assessment is operationalized be **Employee's Knowledge of BPMN**. Employee's knowledge of BPMN indicates the degree to which employees answer a knowledge based test on BPMN.

This section is motivated by the need to assign employees to their groups, and provide training on BPMN modelling. This is to ensure that employees can successfully participate in an experiment based on the matched-pairs design chosen, and that they have the knowledge, skills, and competence to model a business process.

5.3.3.1.1 Matched Subjects

The eighteen employees that took part in the experimental study completed the CRESUS pre-test online questionnaire with results described in Table 5-9. The table described the employees' role, title, level of service, educational qualification, domain knowledge and modelling knowledge. Employees 3940 and 1314 with a title of lecturer grade II and where their role indicates that they are programme directors were assigned the role of business executive. The employees 3738, 1234, 2526, 1920, 2122, 1718, 8910, 1112, 2728, 2324, 2930, 4567, 3132, and 1516 whose titles are lecturer grade I, lecturer grade II, associate faculty, research assistant an computer support tutor from the school of computing are assigned to the role of business users. The employees 3334 and 3536 with the titles senior IT administrator are assigned the role of IT architect.

To aid decision making a ranking of priority was given in the following order to education level, level of service, modelling knowledge and finally domain knowledge. An excellent match occurs if the measures match or are similar by two units on the scales of their measurement. A good match occurs if there is not more than one measure that is disjoint such as course director and staff or not more than one measure that has a difference greater than two units of measurement. An average match occurs if there is not more than one measure that is disjoint and/or not more than two measures have a difference greater than two units. A poor match occurs if there is not more than one measure that is disjoint and/or three or more measures have a difference greater than two units.

The following employees were deemed an excellent match 3940 and 1314; 4567 and 2930; 3132 and 1516; 4567 and 2930.

The following employees were deemed a good match 3132 and 1516; 2526 and 1920; 3738 and 1234; 2324 and 2728. Employees 2324 and 2728 are the same for educational level and domain knowledge. However, there are differences in level of service and modelling knowledge. An observation was that participant 2324 had worked in a similar role prior to NCI and so the difference in level of service was not factored into the judgement for matching the participant therefore this was deemed a good match.

The following employees were deemed an average match 8910 and 1112; 3536 and 3334;

Employees 2122 and 1718 have the same educational level. However there is a sizeable gap between level of service, modelling knowledge and domain knowledge. This was deemed a poor match.

Employees were randomly allocated to the experimental and control groups as follows, employees 3940, 3738, 2526, 1718, 8910, 2728, 2930, 3132 and 3334 to experimental group A. Employees 1314, 1234, 2122, 1920, 1112, 2324, 4567, 1516, and 3536 to control group B;

Table 5-9 Matching Employees Experimental Study 2

Role	Id	Title	Number of years' service with NCI	Number of years' service prior NCI	Level of education attained	Domain Knowledge Rollout of IT Image	Modelling Knowledge BPMN	Other e.g. DFD, UML	Total Modelling	Group
Course Director	3738	Lecturer II	10+	10+	10	1	1	2	3	A
	3940	Lecturer II	7-8	1-2	10	3	3	3	6	A
	1314	Lecturer II	7-8	1-2	10	3	1	5	6	B
	2526	Lecturer II	3-4	10+	9	3	2	4	6	A
	2122	Lecturer II	10+	10+	9	1	1	1	2	B
	8910	Lecturer II	1-2	7-8	9	4	4	4	8	A
	1112	Lecturer I	1-2	1-2	9	3	1	2	3	B
	1718	Associate Lecturer	5-6	10+	9	5	4	5	9	A
Academics / Research Assistant / Support Staff	1234	Associate Lecturer	<1	7-8	10	2	1	4	5	B
	2728	Lecturer I	7-8	0	9	4	1	2	3	A
	1920	Associate Lecturer	3-4	10+	9	3	2	4	6	B
	2324	Associate Lecturer	<1	10+	9	4	4	4	8	B
	4567	Associate Lecturer	<1	10+	8	1	2	3	5	B
	3132	Associate Lecturer	<1	0	9	1	2	4	6	A
	2930	Computer Support Tutor	<1	10+	8	3	1	3	4	A
	1516	Research Assistant	<1	0	9	3	2	3	5	B
IT administrators	3536	Senior IT Administrator	1-2	10+	6	4	4	3	7	B
	3334	Senior IT Administrator	7-8	9-10	7	4	2	2	4	A

5.3.3.1.2 BPMN assessment

The employee's marks from the BPMN assessment were analysed and results show it is not statistically significant ($p <= 0.05$) between experimental group A ($\mu=65$, $\sigma= 21.5$, N=9) and control group B ($\mu=71.33$, $\sigma= 26.981$, N=9), paired t (8) = -0.590, p=0.572. The effect size was low ($\rho = 0.131$). This result demonstrates that there are no significant differences between experimental group A and control group B. Although there is no significant difference between the groups, the means indicate that the control group are marginally higher by 6% on the assessment.

Figure 5-3 indicates that 55% of control group B got over 70% in the assessment compared to 33% of experimental group A.
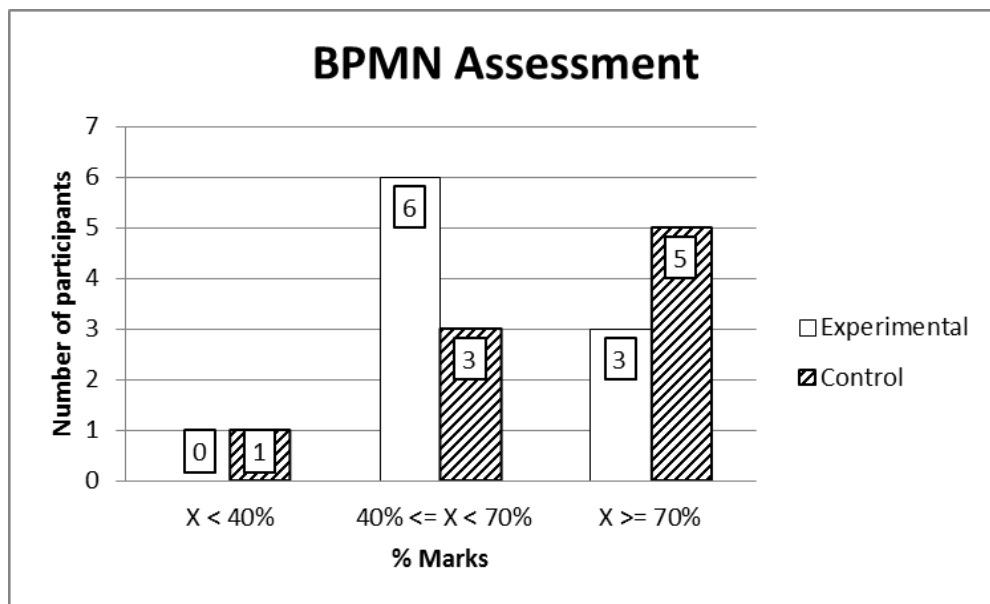
Figure 5-3 Results of BPMN Test

5.3.3.1.3 Discussion and Findings

The results demonstrate that there is no significant difference between the groups in relation to their understanding of business process modelling and notation prior to controlled experiment which rules out modelling bias that may affect the outcome of the experiment. However, the results do indicate that the control group B are marginally higher by 6%. This would indicate that the control group B may perform better than the experimental group A. This is further supported by the results of the assessment, where 22% of the control group where higher than the experimental group in percentage marks above 70%.

5.3.3.2 Collaboration Feature

This section analyses the results of experimental study three in terms of the collaboration feature of the CRESUS framework. Collaboration was operationalized by encompassing one dimension namely ***employee's experience about collaboration during the creation of the data model, IT infrastructure and business process model.***

From Figure 5-4, more than half of experimental group A (56%) had a positive experience when asked in the interview about collaboration during the creation of the data model, IT infrastructure and business process model when CRESUS-T and email were used. The majority of control group B (89%) had a negative experience when email and Google Docs were used. The mean for employees experience with collaboration during the creation of the data model, IT infrastructure and business process model when asked in the interview was a statistically significant ($p <= 0.05$) positive experience when CRESUS-T and email were used ($\mu=0.56$, $\sigma=0.527$, N=9) compared to when email and Google Docs were used ($\mu=0.11$, $\sigma=0.333$, N=9), paired t (8) = 2.530, p=0.035. The effect size was medium ($\rho = 0.316$). This result demonstrates that the CRESUS framework encourages collaborative requirements elicitation.
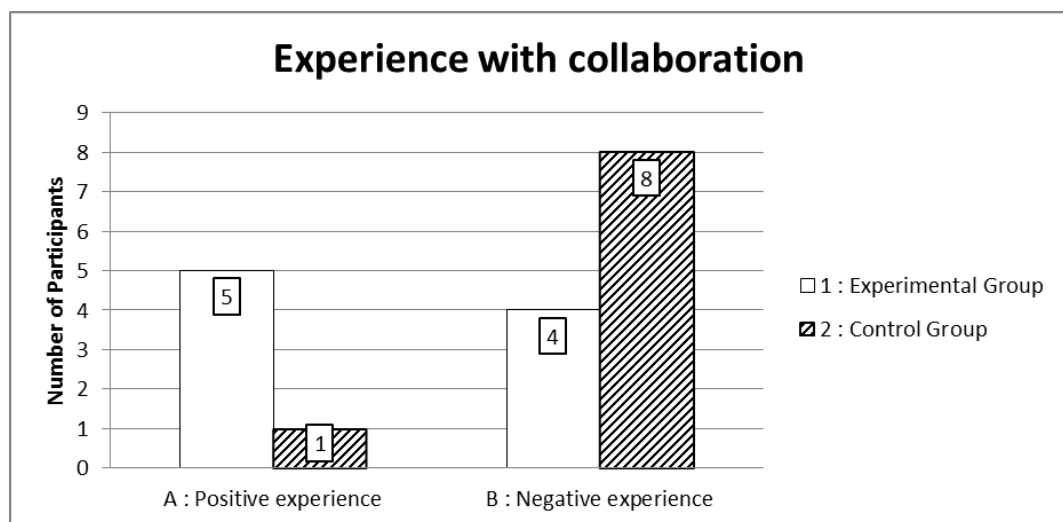
Figure 5-4 Experience with Collaboration

The experimental group expressed in the interview that they liked the collaboration, and found there was a clear and understandable approach to Requirements Elicitation. In relation to collaboration, a sample of one employee's comment "Your on to a good thing …There is more collaboration and more social media where you can give feedback and comment…people are collaborating more … people are starting to blend

things more for example people are sending emails with twitter hash tags and comments and ratings 4 or 5 stars are being used…and an extension of this would be to see how this could fit in the workplace…Use them in email and get people working together get things connected."

The main themes around negative experience for the experimental group related to working independently and participation limited by time.

Other suggestions to improve working collaboratively involved the use of asynchronous communication to help the team move through the different process model activities. One comment from an employee was to use a push technology to remind people if they had to approve requirements or if they had to work on a particular activity.

The main themes around negative experience for the control group that were different from the experimental group relate to limited participation, lack of understanding, bilateral communication and asynchronous communication. Limited participation was affected by illness, work and other commitments, and lack of a shared time that employees can collaborate together. Terminology contributed to the lack of understanding for one employee. There was a perception echoed by several employees that the group fragmented with a lot of bilateral discussions. In addition there was a perception that asynchronous communication does not lend itself to collaboration.

An employee from the experimental group commented during the experiment that for years he used to pass on his software requirements to the IT department and never really understood what was happening. He indicated that through the CRESUS framework he now has a better understanding of what others are doing through collaboration.

5.3.3.3 Discussion and Findings

The comment by another employee around the use of push technology to remind people if they had to approve requirements or if they had to work on a particular activity represents a typical challenge conducting research in a real working environment. Employees were working on their normal duties and didn't realise that they had to complete a collaborative activity that related to completing the Requirements Gathering sub activity of the CRESUS process model before the rest of the group could move on to the next activity. In effect the group were held up. On reflection, this could be

remedied with future research around a collaborative mechanism that interacts with employees during their normal working duties using push technology.

The comment from an employee in relation to collaboration indicates a possible future research direction around collaboration and social media.

The control group appear to have issues around understanding the terminology used by other employees and fragmentation of the group. This demonstrates that the approach with the CRESUS framework of using an ontology-driven process model designed with collaborative activities was successful.

The key finding that arises out of the evaluation of collaboration is that employees had a significant positive experience with collaboration (p=0.35). The experimental group liked the collaboration, and found there was a clear and understandable approach to requirements elicitation, in addition there was visibility and an appreciation of the work other employees do, through collaboration.

5.3.3.4 Communication Feature

This section analyses the results of experimental study three in terms of the communication feature of the CRESUS framework. Communication was operationalized by encompassing two dimensions namely *frequency of communication and employee's experience of asynchronous communication around completing the data model, IT infrastructure and business process model*.

Frequency of communication indicates the degree of messages that employees sent through CRESUS-T, email and Google Docs. Asynchronous communication indicates the degree to which participants have a positive or negative experience of this form of communication based on interviews.

5.3.3.4.1 Frequency of Communication

The following analysis of results is based on the evaluation of logs from the CRESUS-T support tool, email and Google Docs.

The mean frequency of communication was analysed and results show it is statistically significantly ($p \leq 0.05$) higher when CRESUS-T and email are used (μ=14.33, σ= 10.989, N=9) than for the case when email and Google Docs are used (μ=3, σ=2.739, N=9), paired t (8) = 2.802, p=0.023, during the Requirements Elicitation and Modelling

144

and Analysis activities of the CRESUS process model. The effect size was medium ($\rho = -0.316$). This result reinforces the examination of frequency of communication during the Requirements Elicitation activity (cf. section 5.3.2.4.3) which reinforces the proof that the CRESUS framework increases communication. However there are differences in the current experiment such as an additional business process modelling activity in the CRESUS process model for the experimental group and an additional communication channel for the control group.

The logs of CRESUS-T were examined with a refocus on the Requirements Elicitation activity only. The mean frequency of communication metric was further analysed and results show that it is statistically significant ($<= 0.05$) higher when CRESUS-T and email are used ($\mu=12.11$, $\sigma= 2.721$, N=9) compared to the case when email and Google Docs are used ($\mu=2.33$, $\sigma=2.236$, N=9), paired t (8) = 3.365, p=0.010, during the Requirements Elicitation activity of the CRESUS process model. The effect size was large ($\rho = -0.119$, p=0.761). This result augments the results from the examination of frequency of communication during the Requirements Elicitation activity in experimental study two (cf. section 5.3.2.4.3).

The logs of CRESUS-T were examined with a refocus on the Modelling and Analysis activity only. The mean frequency of communication metric was further analysed and results show that it is **not** statistically significant ($<= 0.05$) higher when CRESUS-T and email are used ($\mu=2.22$, $\sigma= 3.492$, N=9) compared to the case when email and Google Docs are used ($\mu=0.67$, $\sigma=1$, N=9), paired t (8) = 1.175, p=0.274, during the Modelling and Analysis activity of the CRESUS process model. The effect size was medium ($\rho = -0.370$, p=0.321). This result implies that the significant difference with frequency of communication arises in the first two activities of the CRESUS process model.

5.3.3.4.2 Discussion and Findings

The finding that frequency of communication did not increase in the Modelling and Analysis activity of the CRESUS process model was initially surprising given the conclusion that frequency of communication significantly increased during the Requirements Elicitation activity. On reflection however, when the employees initially started the experiment, there was a period where they were getting to know the support tool and communication was low. This trend appears to have continued with the move into the Modelling and Analysis activity.

Given that the results of experimental study three reinforce the results of experimental study two, one can conclude that the frequency of communication significantly increased in the Requirements Elicitation activity of the CRESUS process model.

5.3.3.4.3 Experience with Asynchronous Communication

Figure 5-5 shows that more than half the experimental group (67%) had a positive experience when asked in the interview about their experience with asynchronous communication around completing the data model, IT infrastructure model and business process model when CRESUS-T was used. The majority of the control group (89%) had a negative experience when email and Google Docs were used. The mean for employees experience with asynchronous communication around completing the data model, IT infrastructure model and business process model when asked in the interview was a statistically significant ($p <= 0.05$) positive experience when CRESUS-T was used ($\mu=0.67$, $\sigma=0.500$, N=9) compared to when email and Google Docs were used ($\mu=0.11$, $\sigma=0.333$, N=9), paired t (8) = 3.162, p=0.013. The effect size was small ($\rho = 0.250$). This result demonstrates that the CRESUS framework provides a positive experience with asynchronous communication.

The experimental group liked the way people were connected, their requirements were approved, the CRESUS-T tool was intuitive and communication covered most of the issues. An observation was that a participant travelled abroad during the experiment and found it convenient to communicate using asynchronous communication.
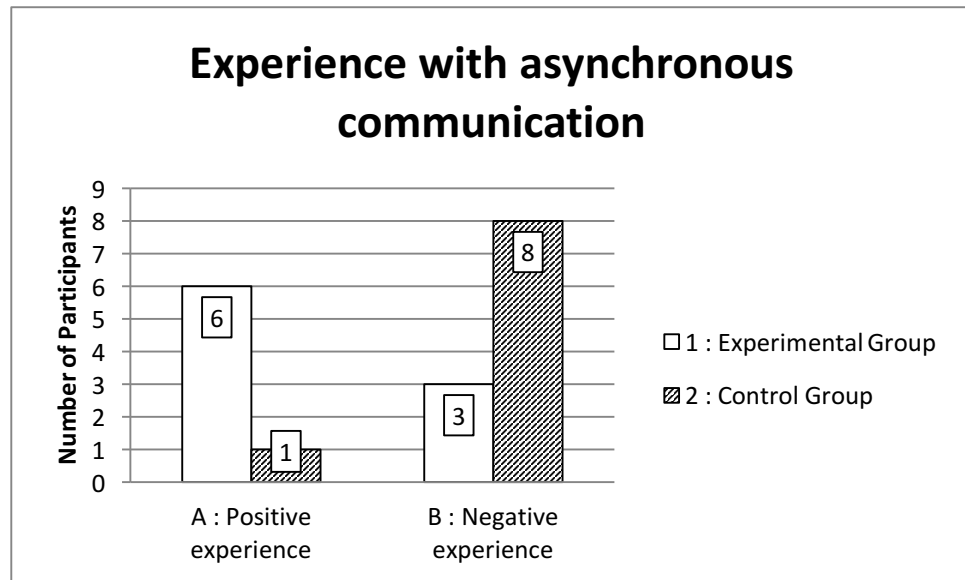
Figure 5-5 Experience with Asynchronous Communication

The predominant issues for the experimental group were around little communication and a lack of understanding of the CRESUS-T support tool which could be addressed with practice.

The issues that were different with the control group centred on trying to communicate asynchronously when there was a mismatch in employees' availability and tasks started to drift without any clear direction.

5.3.3.4.4 Discussion and Findings

Although, employees were requested to work on the experiment every day over the working week for at least 10 minutes between the hours of 10:00 and 12:00, this proved to be an impossible task in a real environment. For both groups, work commitments, flight arrangements and social engagement seemed to pull the participants away from the study. For the control group, asynchronous communication was a real challenge which seemed to frustrate the employees because they had great difficulty identifying a time when everybody was available. The comment around tasks starting to drift without any clear direction seems to indicate that there was a lack of co-ordination or follow up between the employees in completing tasks that involved creating artefacts. This issue did not arise in the CRESUS framework. This highlights the importance of structured collaborative activities that focuses the employee's attention to creating artefacts as they progress through the CRESUS process model.

The results are favourable towards the employees that used the CRESUS framework. This may be because there was increased communication and that the focus and structure of the CRESUS framework kept the employees working on completing collaborative activities.

This seems to suggest that the structure of the CRESUS framework with key knowledge representation artefacts helps to keep the employees focused on the tasks.

5.3.3.5 Prototyping Feature

This section analyses the results of experimental study three in terms of the prototyping feature of the CRESUS framework. Prototyping was operationalized by encompassing one dimensions namely *employee's opinion that the IT infrastructure has the potential to represent an authentic solution.* This dimension indicates the degree to which employees have a positive or negative opinion that the IT infrastructure model has fidelity with commercial or open source software. This metric is based on interviews.

The mean of employees opinion that the IT infrastructure has the potential to represent a real world solution when asked in the interview indicated there was a statistically significant ($p <= 0.05$) positive opinion when CRESUS-T and email were used ($\mu = 1.78$, $\sigma = 0.667$, N=9) than for the case when email and Google Docs was used ($\mu = 0$, $\sigma = 0$, N=9), paired t (8) = 8, p=0.000. This result demonstrates that the automatically generated prototype of the IT infrastructure is realistic.

From Figure 5-6, the majority of the experimental group (89%) that used the CRESUS-T support tool had a positive opinion compared to 100% of the control group that had a negative opinion when asked if the IT infrastructure has the potential to represent a real world solution in the interview.
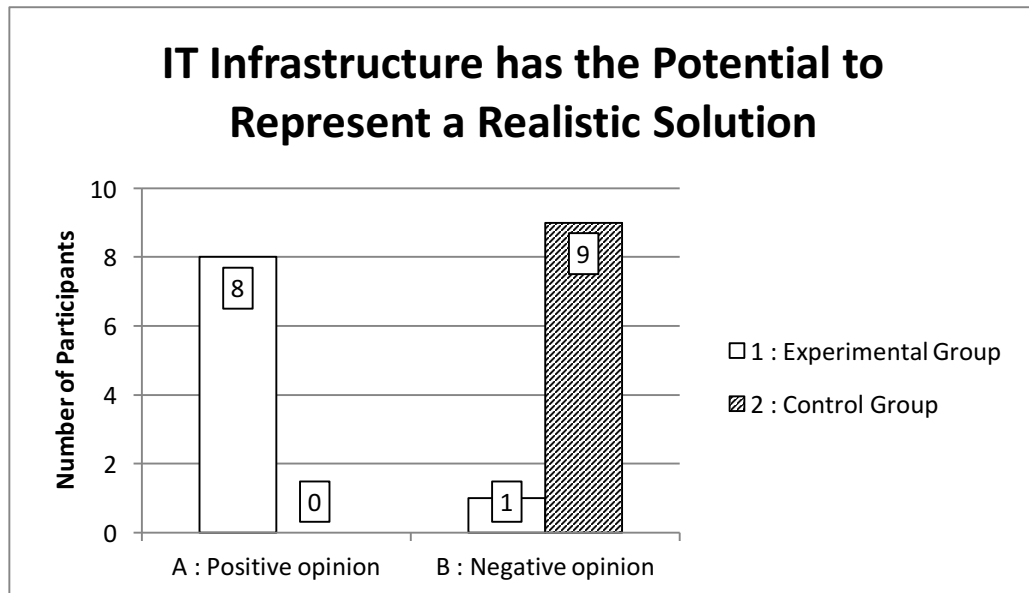
Figure 5-6 Opinion on Fidelity of IT Infrastructure

The experimental group expressed during the interview that they liked the approach that the system automatically generated the IT infrastructure through web service components and deployed them in a business process. They felt it was realistic and allowed people that are located in different geographical areas to collaborate. Samples of comment provided include, "it was more like a dry run and if we were to do it again we could turn out a perfect IT infrastructure. This was more like a sandbox we were still getting our heads around it".

The control group did not create an IT infrastructure however they did create a representation of the IT infrastructure model in Microsoft PowerPoint. One participant logged a journal entry indicating that he was "Making significant progress at building a draft ER model of the imaging infrastructure from the case study". Another employee who represented the IT architect said "From my point of view the IT side I knew exactly in my head I knew exactly what way the model should work. I did not communicate this through email or through interaction with my colleagues."

5.3.3.6 Discussion and Key Findings

The result for the control group is surprising as the employees are technical in nature. In experimental study two it was clear that the IT infrastructure was the part of the study where they spent most of their time. This issue may be linked to the breakdown in asynchronous communication. The comment from the employee that represented the IT architect demonstrates a communication issue that is emerging strongly from control

group B. This really highlights the advantage with CRESUS frameworks design to ensure effective communication as influenced from the state of the art (cf. section 2.3.2).

The key findings that arises out of the evaluation of the prototyping feature is that employees opinion on the potential for the prototype to represent a real world solution was significant (p=0.000). This result reinforces the results in experimental study two, where the majority (66%) of the experimental group that used CRESUS-T indicated that the support tool helped create a shared understanding of how IT infrastructure can support the organisational change. This result is important as it is the basis for aligning the IT infrastructure with the business process. In addition, the state of the art showed that it can lead to the discovery of a richer set of requirements (see section 2.2.3).

5.3.3.7 Modelling Feature

This section analyses the results of experimental study three in terms of the modelling feature of the CRESUS framework. Modelling was operationalized by encompassing three dimensions namely ***Employee's experience of Modelling a Business Process, Model Recall and Problem Solving.***

Experience of modelling a business process indicates the degree to which participants have positive or negative experiences with collaborative business process modelling based on an interview (Appendix 3). Model recall indicates the degree to which participants can produce a business process model for the rollout of the IT software image based on the CRESUS post-test (Appendix 3). Model Understanding indicates the degree which participants can produce a business process model for deploying the IT software image to the classroom by including a scenario that allows the installation of customised software on all the pc's in a specific classroom based on CRESUS post-test (Appendix 3).

5.3.3.7.1 Experience of Business Process Modelling

The following analysis of results is based on the employees's experience with business process modelling based on interviews.

An analysis of the interview results indicated that there was a statistical significant (p<=0.05) relationship in the employees' experience with collaborative business process modelling using the CRESUS-T and email ($\mu=0.78$, $\sigma=0.833$, N=9) compared to when

150

email and Google Docs were used ($\mu=0.00$, $\sigma=0.000$, N=9), paired t (8) = 2.800, p=0.023.

In Figure 5-7, 33% of the experimental group that used the CRESUS-T support tool had a positive experience with collaborative business process modelling, 44% had a negative experience and 22% had a combination of positive and negative experiences. The experimental group expressed in the interview that they liked the model, visibility of the collaboration and problem solving, and an easy to use interface. Samples of comments provided from the experimental group A include: "Good for modelling real world scenarios which illustrates to the other business users (departments) a more complete workflow" and "By using the CRESUS support tool you could see all the work that was done by the other group members, the way they were actually approaching the problem being solved, and how it should interact with that as an employee to make sure that I was actually moving in the right direction."
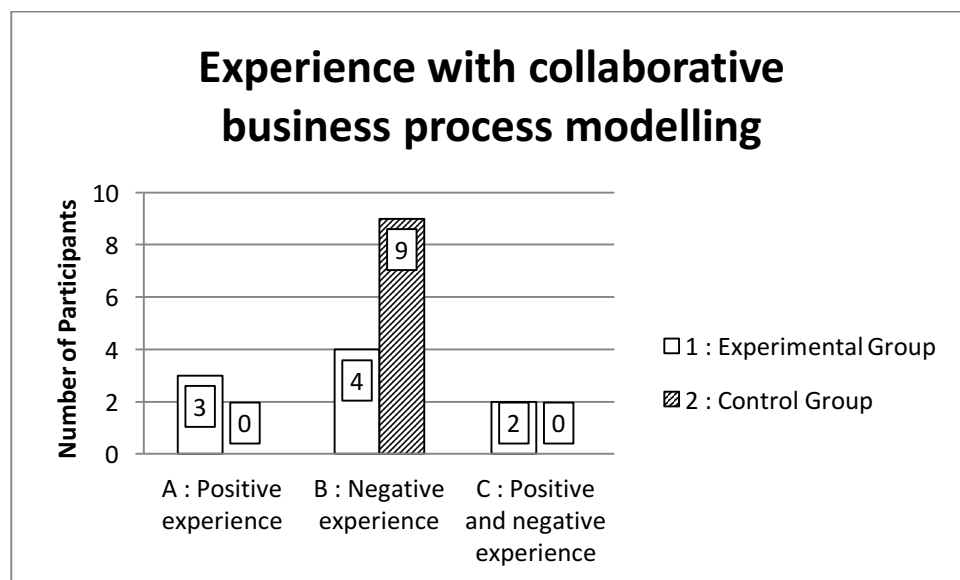
Figure 5-7 Employees Experience with Modelling

The negative themes that emerged from experimental group A, were around modelling independently, not modelling, and requires a specialist role. Two employees felt that they were modelling independently with comments such as "The BPMN I struggled with and I didn't feel that I collaborated very well" and "Everybody acting independently trying to do an individual business process". The primary reason for not modelling in the experimental group was due to travelling abroad or attending a social event (2 employees) and one employee was competing with work pressures and found it difficult to spend time on the experiment. Another employee indicated that the activities

that relate to business process modelling and IT infrastructure required a specialist role. Observations also reinforced comments from the participants in that each employee tried to create a business process from start to finish on their own with no attempt to build on the work of other employees. Journal entries reinforce the general negative themes around business process modelling with one employee from the experimental group recording the following entry on the last day of the experiment "the task demands quite a bit of technical knowledge that I do not have".

100% of the control group had a negative experience with collaborative business process modelling. The control group either did not conduct any modelling or if they did it was not shared with the employees and so collaboration did not take place. The negative themes that emerged from control group B, were around not modelling, difficult to model and modelling independently. A sample of the comments from the control group reinforce the negative theme with a sample comment "it was difficult to define a suitable business process based on the engagement we had" and "I found myself having to do a lot of it on my own."

5.3.3.7.2 Discussion and Findings

Allowing for the effect of travel, attending a social event and not being able to model due to work commitments, then only one employee had a negative experience around modelling independently in the experimental group. The other employee that had a negative experience with modelling independently was part of the group that had a combination of positive and negative experiences. The models that were created during the evaluation were basic and consisted of a start event, sequence flow and a task. Employees spent one day on the business process modelling activity and the remainder of their time on getting to know the system and completing the previous activities of CRESUS process model. This may have contributed to the basic modelling and feeling that it was more independent. Future research may explore optimum configurations for timelines in each process model activity. The results may also indicate that modelling should be carried out by a specialist in the organisation such as a process modeller.

The majority of the control group did not get to the modelling activity. Comments from control group B appear to indicate that poor coordination, lack of engagement, no collaboration, and poor communication were factors in the stakeholder's experience of modelling.

5.3.3.7.3 Model Recall

The following analysis of results is based on the employees's ability to create a business process model of the scenario that they had just completed in this experimental study based on a post-test.

Figure 5-8 shows the frequency distribution of percentage marks, for model recall with the experimental and control groups. An independent samples t-test was conducted on the employees' ability to recall the business process model of the roll out of the IT software image. The mean marks for the employees ability to recall the business process model from the experimental study indicated there was **no** statistically significant ($p <= 0.05$) difference for the experimental group that used CRESUS-T and email ($\mu=54$, $\sigma=23.076$, $N=9$) than for the case when email and Google Docs was used ($\mu=29.22$, $\sigma=27.013$, $N=9$), paired t (8) = 1.635, $p=0.141$. The effect size was large ($\rho = -0.647$). Although there is no significant difference, the mean of the experimental group A is 24.78% higher than control group B.
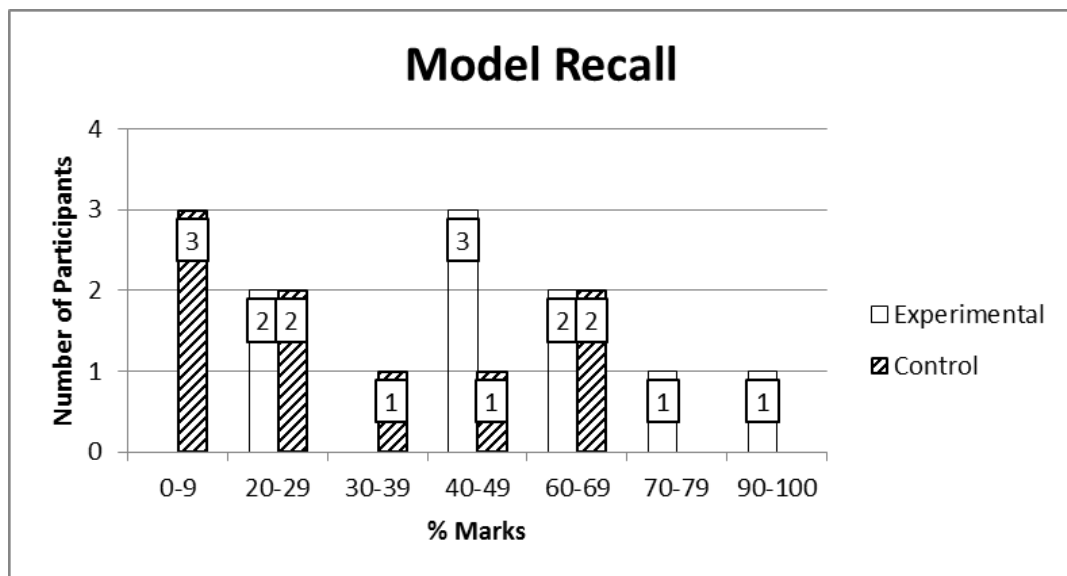
Figure 5-8 Frequency Distribution of Marks for Model Recall

5.3.3.7.4 Discussion and Findings

For model recall it is premised that the higher the model recalls, the higher the level of understanding of the application domain being modelled.

The mean percentage marks for the model recall question in experimental group A was 24.7% higher than the control group B This result correlates with the significant difference in experience between the experimental and control groups with business process modelling (c.f. section 5.3.3.7.1).

From the BPMN assessment (cf. section 5.3.3.1.2), there was an indication that control group B may perform better than the experimental group A. This was based on control group B scoring 22% higher than the experimental group A, in percentage marks above 70%. The initial indicator that the control group was 22% higher on percentage marks in a BPMN assessment and then 24.7% lower in a question on recalling the BPMN model could be attributed to the lack of engagement (c.f. section 5.3.3.7.1).

The experimental group A, created a basic model during the experiment but it was not comparable to the models created in the post test. The answers in the post-test were of much better quality. This may be because the experimental group were learning how to use the tool as they started each of the CRESUS process model activities and were still becoming familiar with CRESUS-T during the Modelling and Analysis activity.

5.3.3.7.5 Problem Solving

The following analysis of results is based on the employees's ability to create a business process model to a different problem from the scenario that they had just completed in this experimental study, based on a post-test.

Figure 5-9 shows the frequency distribution of percentage marks in the test for problem solving with the experimental and control groups. An independent samples t-test was conducted on the results.
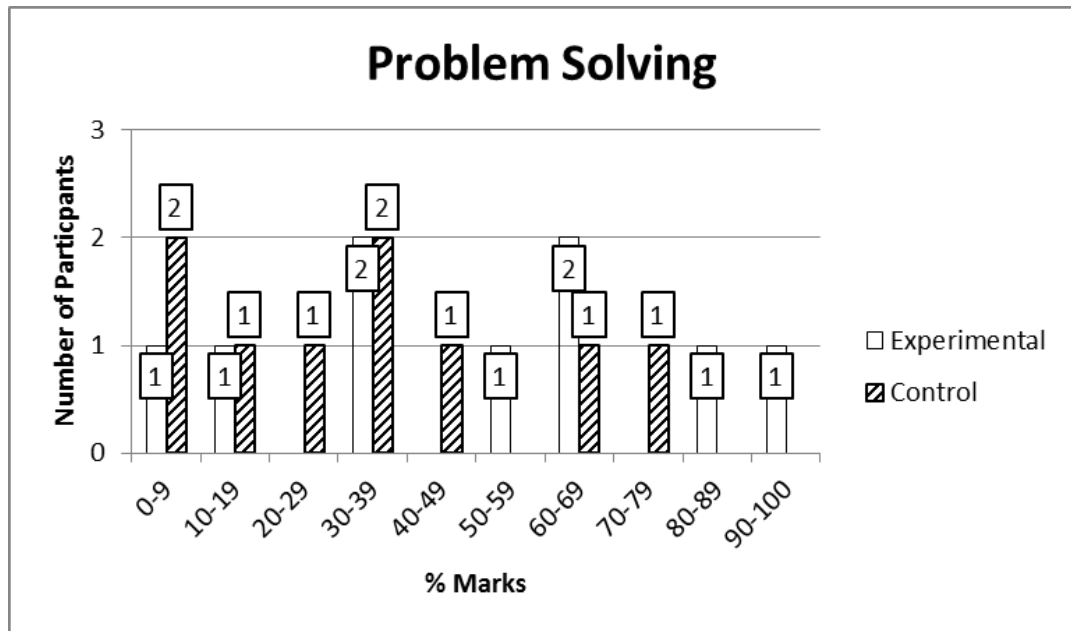
Figure 5-9 Frequency Distribution of Marks for Problem Solving

The mean marks for the employees' ability to answer the problem solving question in the experiment was **not** statistically significant (p<=0.05) for the experimental group that used CRESUS-T and email (μ=47.67, σ=31.906, N=9) than for the case when email and Google Docs was used (μ=33.44, σ=25.812, N=9), paired t (8) = 0.807, p=0.443. The effect size was large (ρ = -0.676).

5.3.3.7.6 *Discussion and Findings*

Problem solving questions require the employees to use the mental understanding that they developed during the experiment which would suggest more sophisticated cognitive understanding of the application domain and the higher is the level of understanding they have developed.

Although here was no significant difference in results, the mean percentage marks of the experimental group A is 14% higher than control group B for problem solving. This would suggest that there is deeper understanding of business process modelling in the experimental group. The experimental group modelled a business process using BPMN. This occurred during the Modelling and Analysis activity of the CRESUS process during the last day of the experiment. The employees were new to BPMN and this may be overwhelming. There are approximately 40 different elements throughout the flow objects, connecting objects, swimlanes and artefacts categories. In addition there are rules about when they elements can and can't be used. Although the there was a series

155

of tutorial on how to use the CRESUS-T support tool and business process modelling, the time taken to get familiar with the tool and BPMN notation may have had an impact on the experimental groups ability to gain a deeper understanding of business process modelling in one day. As such, future research should investigate the appropriate timelines that employees spend in each CRESUS process activity. These timelines should factor in the time for employees to familiarise themselves with the CRESUS-T support tool.

The mean percentage marks for experimental group A with the problem solving question was 47.67. The findings from the evaluation of model recall (cf. section 5.3.3.7.3) showed that the mean percentage marks for experimental group A was 54. This represents a reduction of 6.33%. The result shows that surface knowledge is higher but deeper understanding is reduced. Whereas, in the control group the mean percentage marks for the problem solving question was 33.4 and the mean percentage marks for model recall was 29.2. The control group have increased their mean percentage marks by 4.22%.. One of the employees in the control group interpreted the model recall question, to reproduce the business process diagram that was created during the experiment. As the employee did not create a business process diagram during the experiment, the answer to this question was left blank and the participant got zero marks. For the problem solving question, the employee reproduced a business process diagram and scored 69 marks which represents an overall average 7.6% increase for the control group. Taking this anomaly into account would indicate that overall, the control group would have experienced a reduction in deeper understanding compared to surface knowledge, which is as one would expect.

5.3.3.8 Shared Understanding

Shared understanding indicates the degree to which employees have a positive or negative experience with the approach of reaching a shared understanding during the creation of the data, IT infrastructure and business process model, based on interview.

The mean of employees experience with reaching a shared understanding when asked in the interview indicated there was a statistically significant ($p \leq 0.05$) positive experience when CRESUS-T and email were used ($\mu=1$, $\sigma=1$, $N=9$) compared to when email and google Docs were used ($\mu=0$, $\sigma=0$, $N=9$), paired t (8) = 3, p=0.017. The results indicate that the CRESUS framework fosters shared understanding.

From Figure 5-10, the experimental group that used the CRESUS-T support tool had a 44% more positive experience and 66% less negative experience than the control group when asked about reaching a shared understanding in the interview.
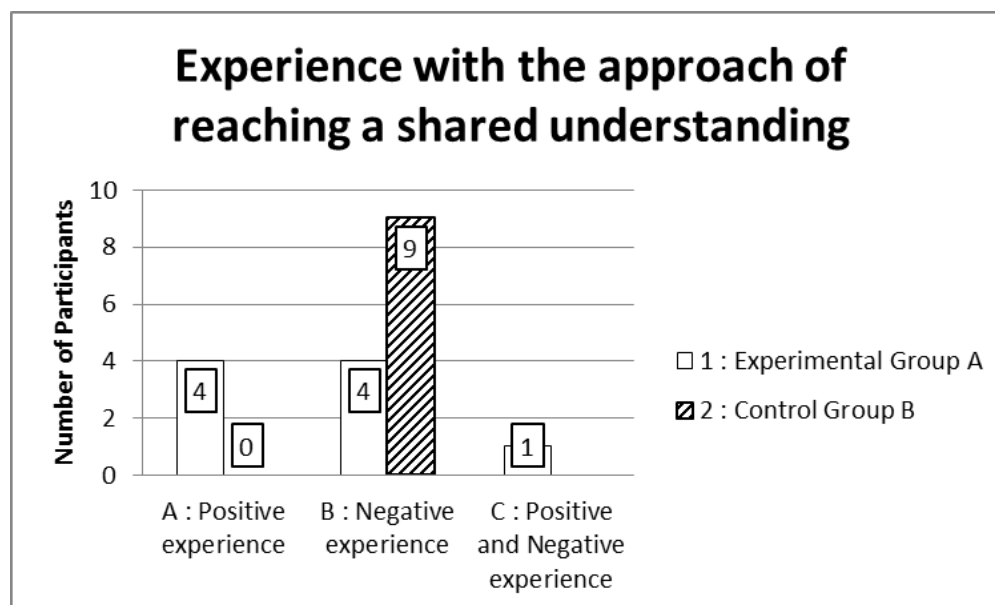
Figure 5-10 Experience with Shared Understanding

Regarding shared understanding, the experimental group liked the collaboration and enhanced understanding, use of the same support tool and modelling language, and use of concrete data with comments like

- "I certainly like the idea of people collaborating and sharing their requirements"
- "The [CRESUS] approach that we followed did enhance the understanding because we were doing it together and you got to know to approve what the requirement the other person has stated".
- "We were all using the same tool same interface same description language so it was easy to understand what everyone was doing."
- "The instance data is useful because it is more concrete to think of relationships in that way. Tackle the problem from two sides. It's going to be more useful for end users to think in terms of the concrete instances that they are associated with and from that you could distil out the more abstract entity relationships."

The negative experience for the experimental group was around competing demands on time, independent understanding, approval process, and use of asynchronous communication. Sample comments provided:-

- "I felt almost isolated so that I didn't feel that I reached a shared understanding or that I shared my understanding with anybody. But we managed to create something in the process".

Issues that arose in the control group centred on a lack of understanding, a lack of coordination, illness, no response from stakeholders through asynchronous communication and work commitments impacting on the stakeholder's ability to participate in the experimental study. Sample of comments provided from control group B are:-

- "Group as a whole didn't work well and come to a shared understanding we didn't have a final shared outcome. Part of the main challenge we had was in the communication where some members were only getting in touch on Thursday which was against my own timetable because I was very busy on Friday. We had some major issues".
- "It was mainly due to work. I was caught up in deadlines".
- "But I never received any direct contact back for my questions".
- "Maybe as associated faculty I felt outside the loop with regards to developing a model here for an IT infrastructure that I wasn't overly familiar".

The artefacts that the control group created during the experiment involved an entity-relationship diagram in word, hand written paper based sketch of a business process, PowerPoint slide of a user data model, and template for identifying instance data such as module requirements. The sketch of the business process was created independently of the group.

5.3.3.9 Discussion and Findings

The employees in control group B had a negative experience with the approach of reaching a shared understanding with comments that indicate issues around collaboration, communication, understanding, illness and work commitments which are representative of a real working environment. The interview brought out an issue around work commitments from one individual, but other employees seemed to take part on an ad-hoc basis also due to work commitments.

The employees in the experimental group were balanced in their experience where some liked the collaboration and others felt isolated. Two of the employees that were

categorised as having a negative experience shared the same office which indicates that there may be some group dynamics. This could be investigated as part of future research. The comment about the use of instance data highlights the advantage with the CRESUS framework in eliciting requirements from the business users' viewpoint where they are familiar with concrete data and from this a requirement engineer could distil the concepts and relationships that a future system requires.

As previously discussed shared understanding can be manifested in the artefacts that are produced. The task of the experiment was to collaboratively create a requirements model, IT infrastructure model and business process model. The control group created several artefacts which were demonstrative of conceptual E-R diagrams or a basic business process diagram. They used a collaborative tool for sharing the models but not the business process diagram. This seems to indicate a training issue around the use of collaborative tools, which could require further investigation.

The experimental group created the requirements model, IT infrastructure model and business process model. Although, there were issues around understanding in the group, they did manage to contribute to creating the artefacts.

The key results that arise from the evaluation of shared understanding is that employees experience with reaching a shared understanding using CRESUS framework was significant ($p=0.017$). This is reinforced by the artefacts created namely the requirements model, IT infrastructure model and business process model. Sample comments provided also support the results such as "The [CRESUS] approach that we followed did enhance the understanding because we were doing it together and you got to know to approve what the requirement the other person has stated" and "We were all using the same tool same interface same description language so it was easy to understand what everyone was doing.".

5.4 Summary

This chapter has discussed the evaluation of the proposed novel CRESUS framework in order to address the research objectives derived from the research question in sections 1.2 and 1.3 of this thesis. Three separate experimental studies were conducted to determine if the CRESUS framework satisfied the objectives of this thesis.

The first experimental study assessed the business executives' desire for a requirements elicitation framework that provided communication and modelling features. Critical

case sampling was used to purposefully select senior and middle managers from institutes of higher education in Ireland as they are the key decision makers to influence change in an organisation. The senior and middle managers were important to this thesis as they yield the most information and have the greatest impact on the development of a requirements elicitation framework.

The second and third experimental studies evaluated the extent support for shared understanding amongst employees was fostered through a collaborative requirements elicitation framework that combined the collaboration, communication, prototyping and modelling features. The evaluations formed a case study for the investigation of the CRESUS framework that supports the alignment of the business process and IT infrastructure in a real environment at NCI. The NCI employees were from the School of Computing and IT department that were directly involved in the change to automate the business process. A randomised matched pairs design with blocking factors was used in the design of the experimental studies. This design removed the effects of the most important nuisance variables and reduced the contaminating effects of the remaining nuisance variables. Such a design led to greater accuracy in the experiment and boosted the power of the experiment with fewer subjects. This allowed accurate deductions to be made regarding the employees experience with the features of the CRESUS framework.

The results presented in this chapter provide confidence that the CRESUS framework supports shared understanding amongst employees through a combination of collaboration, communication, prototyping and modelling. Furthermore this shared understanding is manifested through three knowledge representation artefacts namely a requirements model, IT infrastructure model and a business process model.

6 Conclusion

6.1 Introduction

This thesis has presented the collaborative requirements elicitation framework called CRESUS. The framework consisting of a process model and tool support, introduced a novel way of combining collaboration, communication, prototyping and modelling techniques. The framework may be useful to requirements engineers and business analysts that work on designing and implementing business processes that are aligned to the IT infrastructure.

This chapter discusses the objectives of this thesis and how they were achieved. It also identifies the contribution this work has made to the state of the art of requirements engineering. Finally it concludes with a discussion of future research directions in which this work may be carried forward.

6.2 Objectives and Achievements

The research question posed in chapter on of this thesis was to explore to what extent support for shared understanding amongst employees can be fostered through a collaborative requirements elicitation framework that combines collaboration, communication, prototyping and modelling techniques. Based on this question, four core research objectives were defined.

The first research objective "*to investigate process and techniques which can be used to collaboratively elicit requirements to foster shared understanding*", involved conducting a state of the art review of requirements engineering theory in particular focusing on the role formal knowledge representations have played in the Domain Understanding activity of the CRESUS process model. This included an analysis of requirement elicitation techniques with a focus on combining such techniques such as prototyping and modelling. A specific focus was placed on the prototyping technique as a means to allow the analysis of IT infrastructural choice. Machine translation was used to automatically generate a prototype of the IT infrastructure. Prototyping was an essential feature of CRESUS framework. The literature review also identified modelling of a scenario as an approach for prompting further requirements elicitation. A specific focus was placed on creating enterprise models from web services to form the basis of alignment of the business process to the IT infrastructure. Modelling was an essential

feature of CRESUS framework and it was decided to implement a mapping algorithm that aligns the IT infrastructure components to the model ontology that represents the business process. Collaboration engineering as an approach for the design of collaborative processes was examined. Collaboration was an essential feature of CRESUS framework and it was decided to implement an asynchronous collaborative process model. The literature review also identified the need for a rich channel of communication that embeds techniques for requirements elicitation. Communication was an essential feature of CRESUS framework and it was decided to implement support tool CRESUS-T as a rich channel of asynchronous communication.

In addressing the second objective "*to design a collaborative requirements elicitation framework*", a design that describes a formal process model and support tool was created. The process model comprises of four activities and provides guidance on the order of those activities and the creation of knowledge representation artefacts that employees perform during requirement elicitation. The support tool CRESUS-T was designed around the two central process model activities and incorporated a combination of collaboration, communication, prototyping and modelling techniques.

The third objective "*to implement the framework and produce a supporting tool*" involved implementing CRESUS-T around the two main activities of the CRESUS process model namely Requirements Elicitation and Modelling and Analysis. The main components of the CRESUS-T support tool are around the ontology-driven communication of requirements based on underlying application domain ontology; a transformation engine that automatically generates a prototype representing the IT infrastructure; and a mapping algorithm that aligns the IT infrastructure components to the model ontology.

The fourth objective "*to evaluate the collaborative requirements elicitation framework*" involved evaluating the CRESUS framework around collaboration, communication, prototyping and modelling.

The CRESUS framework was motivated by the desire of 70% of senior and middle managers for a system that would allow them to:-

1. Communicate business goals in natural language.
1. Make use of a support tool that automatically interprets those goals.
2. Identify a business process that can deliver the appropriate services.

The CRESUS framework was evaluated by employees of NCI from the School of Computing and IT department. The evaluation made use of an experimental group that used the CRESUS framework and a control group that used an alternative communication mechanism.

The key findings from the evaluation of the CRESUS framework by NCI employees engaged in the alignment of the business process and IT infrastructure were:-

- Employees had a significant positive experience with collaboration (p=0.035). They liked the collaboration, and found there was a clear and understandable approach to requirements elicitation, in addition there was visibility and an appreciation of the work other employees do, through collaboration. Moreover, they indicated that decisions were made in a collaborative way which was 33% higher than employees that did not use the CRESUS framework. The employees also indicated that their understanding of the IT infrastructure changes to support the organisational goals had increased due to collaboration, and this was 50% higher than the control group.

- Employees experienced significant increased asynchronous communication (p=0.01) during the Requirement Elicitation activity of the CRESUS process model.

- This was further evident by the employees significant positive experience (p=0.013) when asked in the interview about asynchronous communication around completing the data model, IT infrastructure model and business process model when CRESUS-T was used.

- Employees opinion on the potential for the prototype to represent a real world solution was significant (p=0.000).
  - 66% of the employees that used CRESUS-T indicated that the prototype of the IT infrastructure comprising of web services supported the organisational change faithfully

- Employees had a significant positive experience with collaborative business process modelling (p=0.023). A sample comment provided reinforces this positive experience "By using the CRESUS support tool you could see all the work that was done by the other group members, the way they were actually approaching the problem being solved, and how it should interact with that as an employee to make sure that I was actually moving in the right direction."

- Employees demonstrated understanding of the application domain through modelling based on their mean percentage marks for model recall and problem solving. The percentage marks for model recall was 24.7% higher than the control group. The percentage marks for problem solving was 14% higher than the control group.

- Employees experience with reaching a shared understanding was significant (p=0.017). This is reinforced by the three knowledge representation artefacts that were created namely the requirements model, IT infrastructure model and business process model. Sample comments provided also support the results such as "The [CRESUS] approach that we followed did enhance the understanding because we were doing it together and you got to know to approve what the requirement the other person has stated" and "We were all using the same tool same interface same description language so it was easy to understand what everyone was doing."

  o 66% of the employees that used CRESUS-T indicated that the support tool helped them to create a shared understanding of how the IT Infrastructure Model can support the organisational changes.

These results provide confidence that the innovative CRESUS framework fostered shared understanding through a novel combination of collaboration, communication, prototyping and modelling.

6.3 Contribution to the State of the Art

The innovative collaborative requirements elicitation framework, called CRESUS is the major contribution to the State of the Art made by this thesis and the research described therein. This framework consisting of a process model and tool support, introduced a novel way of combining techniques to foster shared understanding. The framework specifically combined collaboration, communication, prototyping and modelling techniques in a novel manner that is significantly different to frameworks used in Requirements Engineering.

The framework includes the CRESUS process model that is formalised and machine understandable. This ontology-driven process model guides the stakeholders on the order of the activities and creation of the knowledge representation artefacts that are machine understandable.

The process model follows the collaborative engineering design approach to creating collaborative patterns that support the stakeholders as they elicit requirements. The novel approach taken here is different from collaborative engineering design approaches in that the collaborative process model has been designed for asynchronous use to support the trend towards global software development.

The novel use of communication involves semantic-mediated communication based on first, an underlying formal knowledge representation of the application domain, and second, a formal knowledge representation of the modelling language. This approach facilitates human and machine understanding through the use of a rich channel for asynchronous communication based on Coughlan & Macredie's (2002) four recommendations for effective communication. The recommendations are for an organization and its stakeholders in attempting to align IT infrastructure. The recommendations are namely, include business users in the design; select an adequate mix of IT architects and business executives who then interact on a collaborative basis; the incorporation of communication activities that relate to knowledge acquisition, knowledge negotiation and user acceptance; the use of elicitation techniques for mediating communication for the requirements of a system such as interviews, group work, prototypes, models and scenarios.

The novel use of prototyping involves the automatic generation of the IT infrastructure based on a code generation process. This code generation process uses machine translation to apply Abbots (1983) and Booch's (1986) textual analysis technique for developing software programs to the vocabulary of a formal knowledge representation, The code generation process derives data types from common nouns, objects from proper nouns and operators (functions, procedures) from verbs. The code generation process combines the data types, objects and operators with the reference architecture to automatically generate the IT infrastructure. The IT infrastructure is automatically deployed to the Cloud using Amazons Web Service.

The novel use of modelling involves creating enterprise models from a formal knowledge representation that through machine translation automatically incorporates the generated IT infrastructure in order to achieve alignment of business process and IT infrastructure.

During the development of the thesis, the research was peer reviewed and published in several international research conferences. The publications which are based on the major contribution of this research in this thesis are:-

- Stynes, P., Conlan, O., O'Sullivan, D. (2009), Supporting Organisational Change through Enhancing Shared Understanding and Simulated Infrastructure Modelling. IN: Proceedings of the PhD Colloquium held at the Winter Simulation Conference, 2009, 13th-16th December, Austin, Texas, USA.

This paper introduces a support tool that aids in the development of a shared understanding in order to support collaborative requirements elicitation to bring about organisational, and associated IT infrastructural, change.

- Stynes, P., Conlan, O., O'Sullivan, D. (2009) Simulation-based Communication Tool: an Enabler for Collaborative Decision Making. IN: Pre-Proceedings of the Workshop on Controlled Natural Language (CNL) 2009, 8th-10th June, Marettimo Island, Italy.

This paper presents the architecture as demonstrated by a support tool that employs a semantically driven natural-language component to capture an organisation's needs. These needs are translated into a simulation, through the use of semantic web services, of a business process that represent the evolution of the organization's IT infrastructure and policies. Through an iterative communication loop between the business executive, IT architect and staff, the aim of the simulation is to accurately represent and specify the IT changes necessary to support the organisation's needs. An initial scenario-based evaluation of this idea that was undertaken among executives in Institutes of higher education in Ireland indicates a desire among executives for such a support tool.

- Stynes, P., Conlan, O., O'Sullivan, D. (2008) Towards a Simulation-based Communication Tool to Support Semantic Business Process Management. IN: Proceedings of the Fourth International Workshop on Semantic Business Process Management in Proceedings of Workshops held at the Fifth European Semantic Web Conference, (ESWC08), 2nd June, Tenerife, Canary Islands, Spain.

This paper proposes a support tool that employs a semantically driven natural-language component to capture a business executive's needs. These needs are translated into a simulation of a business process consisting of semantic web services

that represent the evolution of the organizations IT infrastructure and policies. Through an iterative communication loop with the IT Architect the simulation can be used to accurately represent the changes necessary to meet the business executive's needs.

- Stynes, P. & Conlan, O. (2006) Architectural Framework for the Composition and Delivery of Adaptive Educational Support Services. IN: Proceedings of the International Workshop on Applying Service Oriented Architectures to Adaptive Information Systems in Proceedings of Workshops held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), 20th June, Dublin. Dublin, National College of Ireland. pp. 156-163.

This paper proposes an architectural framework for an integrated adaptive distributed web based education environment. The goal of this framework is to provide a detailed architecture that aggregates heterogeneous systems and provides a container into which systems can plug in and deliver educational services dynamically. Although this paper is focused on web based education environments the architectural approach influences the design around the simulation platform in this thesis.

6.4 Further Research Directions

There are several areas in which there is potential for the research described in this thesis to be extended and advanced. Some of these areas are identified and discussed below and the potential extensions to the support tools developed by this research to address these issues are described.

6.4.1 Effective Communication

Some of the challenges that arose in the evaluation were around the effective use of communication channels (see section 5.3.3.3). Although this related to the use of email which was outside the scope of the CRESUS framework, it does highlight the need to extend the communication mechanism to support modern day communication patterns and disruptive technologies such as social media, messaging applications, instant messaging, and voice. This could include a combination of communication mechanisms such as email incorporating twitter hashtags. These disruptive technologies underlie the need for gathering requirements from the masses. Requirements from the masses is attracting a lot of attention with requirements engineering research practioners and is

the theme for 23rd IEEE International Requirements Engineering Conference 2015 ("RE'15," n.d.). Further research could explore the impact that additional communication mechanism would have on requirements elicitation.

6.4.2 Selection and Combination of Requirement Elicitation Techniques

The state of the art showed that there is a range of requirement elicitation techniques available to support the requirements engineer (see section 2.2.2). The reality is that requirement engineers often select and combine the techniques that they are familiar with. Currently, the CRESUS framework supports the combination of a small library of elicitation techniques that are tightly coupled together. It may be more appropriate to separate the requirement elicitation techniques to give the requirement engineer greater flexibility over the selection and combination of requirement elicitation techniques. Future research could explore this flexibility in an asynchronous setting.

6.4.3 Prototype of the IT infrastructure

The activities of the CRESUS framework should be usable to all stakeholders to facilitate requirements elicitation. Currently, CRESUS automatically generates a prototype of the IT infrastructure that represents the application and data tiers. The CRESUS framework opens a web service tester page that performs the responsibilities of a presentation tier. The presentation tier is rudimentary in design. The automatic generation of the presentation tier that is usable to the business executive would be beneficial, especially in the elicitation of usability requirements. Likewise, the data tier presents raw XML files and could benefit from the generation of a presentation tier that is usable to the business executive to facilitate the elicitation of data requirements.

6.4.4 Alignment of Business Process and IT Infrastructure

The state of the art showed that alignment of business process and IT infrastructure is one of the top issues facing managers (see section 2.3). The alignment issue is around how connected are business and IT strategies. The business executive may want to specify the business strategies in a format other than a business process, such as using the balanced scorecard.

While CRESUS is designed to be flexible in terms of its underlying model ontology there is a need to look at the linkage between different models. For example the use of

the balanced score card to control the business process. If there is a misalignment, then a new set of requirements are dynamically created.

6.5 Final Remarks

I believe that the research presented in this thesis is valuable and brings a significant research contribution to the research community. The CRESUS framework and research findings are useful to both industry and research community.

Requirements Engineering practioners engaging in global software development would benefit from using CRESUS framework to the designing and implementing of business processes. Researchers engaged in investigating the impact that non-functional requirements such as interoperability may have on the IT infrastructure would find the CRESUS framework beneficial.

BIBLIOGRAPHY

(ACE Editor, 2011)   ACE Editor. (2011). Available online at http://attempto.ifi.uzh.ch/site/resources/.

(AceWiki v0.5.1, 2011)   AceWiki v0.5.1. (2011). Available online at https://github.com/AceWiki/AceWiki.

(Aho, Lam, Sethi, & Ullman, 2006)   Aho, A. Lam, M. Sethi, R. & Ullman, J. (2006). *Compilers: Principles, Techniques, and Tools*, 2/E. Prentice Hall. Available online at http://www.pearsonhighered.com/educator/product/Compilers-Principles-Techniques-and-Tools/9780321486813.page.

(Al Balushi, Sampaio, & Loucopoulos, 2013)   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(2), 129–151. http://doi.org/10.1111/j.1468-0394.2012.00625.x.

(Alexander, & Maiden, 2005)   Alexander, I. F. & Maiden, N. (2005). *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*. John Wiley & Sons.

(Allemang, 2008)   Allemang, D. (2008). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL (1 edition.)*. Amsterdam ; Boston: Morgan Kaufmann.

(Amazon Web Services, n.d.)   Amazon Web Services. (n.d.). Available online at http://aws.amazon.com/.

(Ameller, Ayala, Cabot, & Franch, X2013)   Ameller, D. Ayala, C. Cabot, J. & Franch, X. (2013). *Non-functional Requirements in Architectural Decision Making.* IEEE Software, 30(2), 61–67. doi:10.1109/MS.2012.176.

(Antonucci, Bariff, Benedict, Champlin, Downing, Franzen, Madison, Lusk, Spanyi, Treat, Zhao, & Raschke, 2009)   Antonucci, Y. L. Bariff, M. Benedict, T. Champlin, B. Downing, B. Franzen, J. Madison, D. Lusk, S. Spanyi, A. Treat, M. Zhao, L. Raschke, R. (2009). *Business Process Management Common Body Of Knowledge*. Chicago, Ill.: CreateSpace Independent Publishing Platform.

(Anwer & Ikram, 2008)   Anwer, S. & Ikram, N. (2008). *A Process for Goal Oriented Requirement Engineering.* In Proceedings of the IASTED International Conference on Software Engineering (pp. 255–261). Anaheim, CA, USA: ACTA Press. Available online at http://dl.acm.org/citation.cfm?id=1722603.1722649.

(Apache Ant v1.8.2., 2010)   Apache Ant v1.8.2. (2010). Available online at http://ant.apache.org/.

(Apache Jena v2.6.4., 2010)	Apache Jena v2.6.4. (2010). Available online at https://jena.apache.org/.
(Asur & Hufnagel, 1993)	Asur, S. & Hufnagel, S. (1993). *Taxonomy of rapid-prototyping methods and tools.* In, Fourth International Workshop on Rapid System Prototyping, 1993. Shortening the Path from Specification to Prototype. Proceedings (pp. 42–56). doi:10.1109/IWRSP.1993.263196.
(Aurum & Wohlin, 2005)	Aurum, A. & Wohlin, C. (2005). *Engineering and Managing Software Requirements.* Springer Science & Business Media.
(Azadegan, Cheng, Niederman, & Yin, 2013)	Azadegan, A. Cheng, X. Niederman, F. & Yin, G. (2013). *Collaborative Requirements Elicitation in Facilitated Collaboration: Report from a Case Study.* In 2013 46th Hawaii International Conference on System Sciences (HICSS) (pp. 569–578). doi:10.1109/HICSS.2013.136.
(Boehm & Ross, 1989)	Boehm, B.W. and Ross, R. (1989). *Theory W Software Project Management: Principles and Examples.* IEEE Trans. Software Engr.
(Boehm1988)	Boehm, B.W. (1988). A Spiral Model of Software Development and Enhancement. Commiter, pp. 61-72.
(Bagheri & Ghorbani, 2008)	Bagheri, E. & Ghorbani, A. A. (2008). *Experiences on the Belief-Theoretic Integration of Para-consistent Conceptual Models.* In 19th Australian Conference on Software Engineering, 2008. ASWEC 2008 (pp. 357–366). http://doi.org/10.1109/ASWEC.2008.4483224.
(Bailey, 2008)	Bailey, R. A. (2008). *Design of Comparative Experiments.* Cambridge University Press. Available online at http://www.cambridge.org/ie/academic/subjects/statistics-probability/statistical-theory-and-methods/design-comparative-experiments?format=HB.
(Bajpai, & Gorthi, 2012).	Bajpai, V. & Gorthi, R. P. (2012). *On non-functional requirements: A survey.* In 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1–4). doi:10.1109/SCEECS.2012.6184810.
(Bano & Zowghi, 2013).	Bano, M. & Zowghi, D. (2013). *Users' involvement in requirements engineering and system success.* In 2013 IEEE Third International Workshop on Empirical Requirements Engineering (EmpiRE) (pp. 24–31). doi:10.1109/EmpiRE.2013.6615212.
(Bass, Clements & Kazman, 2003)	Bass, L. Clements, P. & Kazman, R. (2003). *Software Architecture in Practice.* (2 edition.). Boston: Addison-Wesley Professional.
(Baxter & Sommerville, 2011)	Baxter, G. & Sommerville, I. (2011). *Socio-technical systems: From design methods to systems engineering.* Interacting with Computers, 23(1), 4–17.

(Berglund, 2006)   Berglund, A. (2006). *Extensible Stylesheet Language (XSL) Version 1.1.* Available online at http://www.w3.org/TR/xsl11/.

(Berners-Lee, Hendler & Lassila, 2001)   Berners-Lee, T. Hendler, J. & Lassila, O. (2001). *The Semantic Web.* Scientific American, 284, 34–43. doi:10.1038/scientificamerican0501-34.

(Bernstein & Kaufmann, 2006)   Bernstein, A. & Kaufmann, E. (2006). *GINO-A Guided Input Natural Language Ontology Editor.* IN: Proceedings of the 5th International Semantic Web Conference. (ISWC 2006), 5-9, November, Athens, USA. Berlin, Springer. pp. 144-157.

(Bernstein, Kaufmann, Göhring & Kiefer, 2005)   Bernstein, A. Kaufmann, E. Göhring, A. & Kiefer, C. (2005). *Querying Ontologies: A Controlled English Interface for End-Users.* IN: Proceedings of the 4th International Semantic Web Conference (ISWC 2005), 6-10 November, Galway, Ireland. Berlin, Springer. pp 112-126.

(Berzins, Martell, Luqi & Adams, 2008)   Berzins, V. Martell, C. Luqi, & Adams, P. (2008). *Innovations in Natural Language Document Processing for Requirements Engineering.* In B. Paech & C. Martell (Eds.), Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs (pp. 125–146). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/978-3-540-89778-1_11.

(Beyer & Holtzblatt, 1998)   Beyer, H. & Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centered Systems.* Morgan Kaufmann Publishers. San Francisco, CA.

(Bittner & Leimeister, 2013)   Bittner, E. A. & Leimeister, J. M. (2013). *Why Shared Understanding Matters – Engineering a Collaboration Process for Shared Understanding to Improve Collaboration Effectiveness in Heterogeneous Teams.* In 2013 46th Hawaii International Conference on System Sciences (HICSS) (pp. 106–114). doi:10.1109/HICSS.2013.608.

(Bleistein, Cox & Verner, 2006)   Bleistein, S. J. Cox, K. & Verner, J. (2006). *Validating Strategic Alignment of Organizational IT Requirements Using Goal Modeling and Problem Diagrams.* J. Syst. Softw. 79(3), 362–378. doi:10.1016/j.jss.2005.04.033.

(Bødker, Kensing & Simonsen, 2011)   Bødker, K. Kensing, F. & Simonsen, J. (2011). *Participatory Design in Information Systems Development.* In H. Isomäki & S. Pekkola (Eds.), Reframing Humans in Information Systems Development (pp. 115–134). Springer London. Available online at http://link.springer.com/chapter/10.1007/978-1-84996-347-3_7.

(Bødker, Ehn, Knudsen, Kyng & Madsen, 1988)	Bødker, S. Ehn, P. Knudsen, J. Kyng, M. & Madsen, K. (1988). *Computer support for cooperative design*. In Proceedings of the 1988 ACM conference on Computer-supported cooperative work (pp. 377–394). ACM. Available online at http://dl.acm.org/citation.cfm?id=62296.
(Boehm, 2006)	Boehm, B. (2006). *A view of 20th and 21st century software engineering*. In Proceedings of the 28th international conference on Software engineering (pp. 12–29). ACM. Available online at http://dl.acm.org/citation.cfm?id=1134288.
(Boehm, 2006)	Boehm, B. (2006). *Some future trends and implications for systems and software engineering processes. Systems Engineering*, *9*(1), 1–19.
(Boehm, 1988)	Boehm, B. W. (1988). *A spiral model of software development and enhancement. Computer*, *21*(5), 61–72.
(Boehm, Bose, Horowitz & Lee, 1994)	Boehm, B. Bose, P. Horowitz, E. & Lee, M. J. (1994). *Software requirements as negotiated win conditions*. In Proceedings of ICRE (pp. 74–83).
(Booch, 1986)	Booch, G. (1986). *Object-oriented development. Software Engineering, IEEE Transactions on*, (2), 211–221.
(Boose, 1989)	Boose, J. H. (1989). *A survey of knowledge acquisition techniques and tools. Knowledge Acquisition*, *1*(1), 3–37.
(Borg & Gall, 1989)	Borg, W. R. & Gall, M. D. (1989). *Educational Research: An Introduction*. (5 ed.). New York: Longman.
(Borgida, Greenspan & Mylopoulos, 1985)	Borgida, A. Greenspan, S. & Mylopoulos, J. (1985). *Knowledge Representation as the Basis for Requirements Specifications*. Computer, 18(4), 82–91. doi:10.1109/MC.1985.1662870.
(Bostrum, 1989)	Bostrum RP. (1989). *Successful application of communication techniques to improve the systems development process*. Inform Manage 1989; 16: 279-295.
(Bourque, & Fairley, (eds.), 2014)	Bourque, P. & Fairley, R. E. eds. (2014). *Guide to the Software Engineering Body of Knowledge, Version 3.0.* IEEE Computer Society, 2014. September 11, 2014, Available online at https://www.createspace.com/4581027?ref=1147694&utm_id=6026
(BPMN Introduction, 2013)	*BPMN Introduction. (2013). Available online at https://www.youtube.com/watch?v=JM-mw0aMNak&feature=youtube_gdata_player.*

(Bray, Paoli, Sperberg-McQueen, Maler & Yergeau, 2008)	Bray, T. Paoli, J. Sperberg-McQueen, C. M. Maler, E. & Yergeau, F. (2008). *Extensible Markup Language (XML) 1.0 (Fifth Edition).* Available online at http://www.w3.org/TR/2008/REC-xml-20081126/.
(Briggs, De Vreede & Nunamaker Jr, 2003)	Briggs, R. O. De Vreede, G.-J. & Nunamaker Jr, J. (2003). *Collaboration engineering with ThinkLets to pursue sustained success with group support systems.* J. of Management Information Systems, 19(4), 31–64.
(Briggs, Kolfschoten, Gert-Jan & Douglas, 2006)	Briggs, R. Kolfschoten, G. Gert-Jan, V. & Douglas, D. (2006). *Defining key concepts for collaboration engineering.* AMCIS 2006 Proceedings, 17.
(Browne & Rogich, 2001)	Browne, G. and Rogich, M. (2001). *An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques.* Journal of Management Information Systems, 17, 4 (Spring 2001), pp. 223-249.
(Bruegge & Dutoit, 2009)	Bruegge, B. & Dutoit, A. H. (2009). *Object-Oriented Software Engineering Using UML, Patterns, and Java.* (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.
(Budde, Kautz, Kuhlenkamp & Züllighoven, 1992)	Budde, R. Kautz, K. Kuhlenkamp, K. & Züllighoven, H. (1992). *Prototyping.* In Prototyping (pp. 33–46). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/978-3-642-76820-0_5.
(Bulles, Freund, Gagne, Menge, Kloppmann, Nijssen, Navarro-Suarez, Trickovic & White, 2010)	Bulles, J. Freund, J. Gagne, D. Menge, F. Kloppmann, M. Nijssen, S. Navarro-Suarez, G. Trickovic, I. White, S. (2010), *BPMN 2.0 by Example Version 1.0.* June 19, 2014, Available online at http://www.omg.org/cgi-bin/doc?dtc/10-06-02.
(Campbell, 2005)	Campbell, B. (2005). *Alignment: Resolving ambiguity within bounded choices.* Available online at http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1166&context=pacis2005.
(Campbell & Stanley, 1963)	Campbell, D. T. & Stanley, J. C. (1963). *Experimental and quasi-experimental designs for research on teaching.* In N.L. Gagge (Ed.) Handbook of research on teaching. Chicago: Rand McNally. (Reprinted as Experimental and quasi-experimental design for research. Chicago: Rand McNally, 1966).
(Camptasisa Studio, n.d.)	Camptasisa Studio. Available online at http://www.techsmith.com/camtasia.html.

(Carmel, 1999)	Carmel, E. (1999). *Global software teams*. Upper Saddle River, NJ: Prentice Hall. Available online at http://assets.conferencespot.org/fileserver/file/120530/filename/Perspectives_2002_38.pdf.
(Castañeda, Ballejos & Caliusco, 2012)	Castañeda, V. Ballejos, L. C. & Caliusco, M. L. (2012). *Improving the Quality of Software Requirements Specifications with Semantic Web Technologies*. In WER. Available online at http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER12/paper_4.pdf.
(Castañeda, Ballejos, Caliusco & Galli, 2010)	Castañeda, V. Ballejos, L. Caliusco, M. L. & Galli, M. R. (2010). *The use of ontologies in requirements engineering*. Global Journal of Researches in Engineering, 10(6). Available online at http://engineeringresearch.org/index.php/GJRE/article/view/76.
(Castañeda, Ballejos, Caliusco & Galli, 2010)	Castañeda, V. Ballejos, L. Caliusco, M. L. & Galli, M. R. (2010). *The Use of Ontologies in Requirements Engineering*. Global Journal of Researches In Engineering, 10(6). Available online at http://www.engineeringresearch.org/index.php/GJRE/article/view/76.
(Ceri, & Fraternali, 1997)	Ceri, S. & Fraternali, P. (1997). *Designing Database Applications with Objects and Rules: The IDEA Methodology*. Harlow: Addison Wesley.
(Chan & Reich, 2007)	Chan, Y. E. & Reich, B. H. (2007). *IT alignment: what have we learned?*. Journal of Information Technology, 22(4), 297–315.
(Chandrasekaran, Silver, Miller, Cardoso & Sheth, 2002)	Chandrasekaran, S. Silver, G. Miller, J. Cardoso, J. & Sheth, A. (2002). *Web service technologies and their synergy with simulation*. IN: Proceedings of the 34th Winter Simulation Conference: exploring new frontiers (WSC 2002), 8-11, December, San Diego, California, USA. Winter Simulation Conference. pp 606-615.
(Checkland, 1989)	Checkland P. (1989). *Soft systems methodology: rational analysis for a problematic world*. John Wiley & Sons, NY, 1989.
(Chin, Diehl & Norman, 1988)	Chin, J.P. Diehl, V.A. Norman, K.L. (1988). *Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface*. ACM CHI'88 Proceedings, 213-218. ©1988 ACM. http://hcibib.org/perlman/question.cgi?form=QUIS.
(Christensen, Curbera, Meredith & Weerawarana, 2001)	Christensen, E. Curbera, F. Meredith, G. & Weerawarana, S. (2001). *Web Service Definition Language (WSDL) 1.1*. Available online at http://www.w3.org/TR/wsdl.

175

(Clark, Harrison, Jenkins, Thompson & Wojcik, 2005)	Clark, P. Harrison, P. Jenkins, T. Thompson, J. A. & Wojcik, R. H. (2005). *Acquiring and Using World Knowledge Using a Restricted Subset of English*. In FLAIRS Conference (pp. 506–511). Available online at http://www.aaai.org/Papers/FLAIRS/2005/Flairs05-083.pdf.
(Cleland-Huang, 2013)	Cleland-Huang, J. (2013). *Meet Elaine: A Persona-Driven Approach to Exploring Architecturally Significant Requirements*. IEEE Software, *30*(4), 18–21. doi:10.1109/MS.2013.80
(Leondes & Jackson, 1992)	Leondes, C.T. Jackson, R.H.F. (1992). *Manufacturing and Automation Systems: Techniques and Technologies*. Academic Press, ISBN 0-12-012745-8, pp.97.
(Coughlan & Macredie, 2002)	Coughlan, J. & Macredie, R. D. (2002). *Effective communication in requirements elicitation: A comparison of methodologies*. Requirements Engineering, 7(2), 47–60.
(Coughlan, Lycett & Macredie, 2003)	Coughlan, J. Lycett, M. & Macredie, R. D. (2003). *Communication issues in requirements elicitation: a content analysis of stakeholder experiences*. Information and Software Technology, 45(8), 525–537. doi:10.1016/S0950-5849(03)00032-6.
(Cregan, Schwitter & Meyer, 2007)	Cregan, A. Schwitter, R. & Meyer, T. (2007). *Sydney OWL Syntax-towards a Controlled Natural Language Syntax for OWL 1.1*. In OWLED (Vol. 258).
(CRESUS Approach BPMN Model Part 4, 2013)	CRESUS Approach BPMN Model Part 4. (2013). *Available online at https://www.youtube.com/watch?v=tpvgQCZJChg&feature=youtube_gdata_player.*
(CRESUS Approach IT Infrastructure Part 3, 2013).	CRESUS Approach IT Infrastructure Part 3. (2013). Available online at https://www.youtube.com/watch?v=iPDDvTIRLyY&feature=youtube_gdata_player.
(CRESUS Approach Part 1, 2013)	CRESUS Approach Part 1. (2013). Available online at https://www.youtube.com/watch?v=uMnfl-bF10k&feature=youtube_gdata_player.
(CRESUS-T Data Entry Part 2, 2013).	CRESUS-T Data Entry Part 2. (2013). Available online at https://www.youtube.com/watch?v=uiJMEZNXTfY&feature=youtube_gdata_player.
(Creswell, 2008)	Creswell, J. W. (2008). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (3rd edition). Thousand Oaks, Calif: SAGE Publications, Inc.
(Curtis, Kellner & Over, 1992)	Curtis, B. Kellner, M. and Over, J. (1992). *Process Modeling*. Communication of the ACM, Vol. 35, No.9, 1992.

(Cycyota & Harrison, 2006)	Cycyota, C. S. & Harrison, D. A. (2006). *What (Not) to Expect When Surveying Executives A Meta-Analysis of Top Manager Response Rates and Techniques Over Time*. Organizational Research Methods, 9(2), 133–160. doi:10.1177/1094428105280770.
(Preston & Karahanna, 2004)	Preston, D. & Karahanna, E. (2004). *Mechanisms for the development of shared mental models between the CIO and top management team*, in Proc. 25th Annu. Int. Conf. Inf. Syst. pp. 465–480.
(Damian & Zowghi, 2002)	Damian, D. E. & Zowghi, D. (2002). *The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization*. In IEEE Joint International Conference on Requirements Engineering, 2002. Proceedings (pp. 319–328). doi:10.1109/ICRE.2002.1048545.
(Damian, Eberlein, Shaw & Gaines, 2003)	Damian, D. E. Eberlein, A. Shaw, M. L. G. & Gaines, B. R. (2003). *An exploratory study of facilitation in distributed requirements engineering*. Requirements Engineering, 8(1), 23–41. doi:10.1007/s00766-002-0164-7.
(Dardenne, van Lamsweerde & Fickas, 1993)	Dardenne, A. van Lamsweerde, A. & Fickas, S. (1993). *Goal-directed requirements acquisition*. Science of Computer Programming, 20(1-2), 3–50.
(Davis, Dieste, Hickey, Juristo & Moreno, 2006)	Davis, A. Dieste, O. Hickey, A. Juristo, N. & Moreno, A. M. (2006). *Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review*. In Requirements Engineering, 14th IEEE International Conference (pp. 179–188). doi:10.1109/RE.2006.17.
(Davis, 1989)	Davis, F. D. (1989). *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technolo*gy. MIS Quarterly, *13*(3), 319–340. http://doi.org/10.2307/249008
(De Vreede, Briggs & Massey, 2009)	De Vreede, G. J. Briggs, R. O. & Massey, A. P. (2009). *Collaboration engineering: Foundations and opportunities:* Editorial to the special issue on the journal of the association of information systems. Journal of the Association for Information Systems, 10(3), 7.
(De Vreede & Briggs, 2005)	De Vreede, G. & Briggs, R. O. (2005). *Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks*. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05 (p. 17c–17c). doi:10.1109/HICSS.2005.144.
(De Vreede, Kolfschoten & Briggs, 2006)	De Vreede, G.-J. Kolfschoten, G. L. & Briggs, R. O. (2006). *ThinkLets: a collaboration engineering pattern language*. International Journal of Computer Applications in Technology, 25(2), 140–154.

(Dean, Lee, Pendergast, Hickey & Nunamaker Jr, 1997)	Dean, D. L. Lee, J. D. Pendergast, M. O. Hickey, A. M. & Nunamaker Jr, J. F. (1997). *Enabling the effective involvement of multiple users: methods and tools for collaborative software engineering*. Journal of Management Information Systems, 179–222.
(Dean, Lee, Pendergast, Hickey & Nunamaker Jr., 1998)	Dean, D. Lee, J. Pendergast, M. O. Hickey, A. M. Nunamaker Jr. J. F. (1998). *Enabling the Effective Involvement of Multiple Users: Methods and Tools for Collaborative Software Engineering*. Journal of Management Information Systems, 14, 3, pp. 179-222.
(Dermeval, Vilela, Bittencourt, Castro, Isotani, Brito & Silva, 2015)	Dermeval, D. Vilela, J. Bittencourt, I. I. Castro, J. Isotani, S. Brito, P. & Silva, A. (2015). *Applications of ontologies in requirements engineering: a systematic review of the literature*. Requirements Engineering. http://doi.org/10.1007/s00766-015-0222-6.
(Dobson & Sawyer, 2006)	Dobson, G. & Sawyer, P. (2006). *Revisiting ontology-based requirements engineering in the age of the semantic web*. In Proceedings of the International Seminar on Dependable Requirements Engineering of Computerised Systems at NPPs. Available online at http://nacho.cps.unizar.es/Docencia/MasterUPV/Articulos/Revisiting%20ontology-based%20RE.pdf.
(Dolbear, Hart, Kovacs, Goodwin & Zhou, 2007)	Dolbear, C. Hart, G. Kovacs, K. Goodwin, J. & Zhou, S. (2007). *The Rabbit language: description, syntax and conversion to OWL*. Ordnance Survey Research Labs Technical Report.
(Echo Web Framework v2.2.1., 2009)	Echo Web Framework v2.2.1. (2009). Available online at http://echo.nextapp.com/site/echo2.
(Evaristo, Watson-Manheim, & Audy, 2005)	Evaristo, R. Watson-Manheim, M. B. & Audy, J. (2005). *e-Collaboration in Distributed Requirements Determination*: International Journal of E-Collaboration, 1(2), 40–56. doi:10.4018/jec.2005040103.
(eXistdb - The Open Source Native XML Database, n.d.)	eXistdb - The Open Source Native XML Database. (n.d.). Available online at http://exist-db.org/exist/apps/homepage/index.html.
(Finkelstein, Kramer, Nuseibeh, Finkelstein & Goedicke, 1992)	Finkelstein, A. Kramer, J. Nuseibeh, B. Finkelstein, L. & Goedicke, M. (1992). *Viewpoints: A framework for integrating multiple perspectives in system development*. International Journal of Software Engineering and Knowledge Engineering, 02(01), 31–57. doi:10.1142/S0218194092000038.

(Floyd, Mehl, Reisin, Schmidt & Wolf, 1989)	Floyd, C. Mehl, W.-M. Reisin, F.-M. Schmidt, G. & Wolf, G. (1989). *Out of Scandinavia: Alternative approaches to software design and system development*. Human-Computer Interaction, 4(4), 253–350.
(Flynn & Jazi, 1998)	Flynn D.J., Jazi M.D. (1998). *Constructing user requirements: a social process for a social context*. Information Systems Journal 1998; 8 (1): 53-83.
(Foster, 2014)	Foster, E. (2014). *Software Engineering: A Methodical Approach* (2014 edition). Apress.
(Fuchs, Höfler, Kaljurand, Rinaldi & Schneider, 2005)	Fuchs, N. E. Höfler, S. Kaljurand, K. Rinaldi, F. & Schneider, G. (2005). *Attempto controlled english: A knowledge representation language readable by humans and machines*. In Reasoning Web (pp. 213–250). Springer. Available online at http://link.springer.com/chapter/10.1007/11526988_6.
(Fuchs, Kaljurand, & Kuhn, 2008)	Fuchs, N. E. Kaljurand, K. and Kuhn, T. (2008). *Attempto Controlled English for Knowledge Representation*. In Cristina Baroglio, Piero A. Bonatti, Jan Maluszynski, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert, editors, Reasoning Web, Fourth International Summer School 2008, number 5224 in Lecture Notes in Computer Science, pages 104–124. Springer, 2008.
(Funk, Tablan, Bontcheva, Cunningham, Davis & Handschuh, 2007)	Funk, A. Tablan, V. Bontcheva, K. Cunningham, H. Davis, B. & Handschuh, S. (2007). *CLOnE: Controlled Language for Ontology Editing*. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, P. Cudré-Mauroux (Eds.), The Semantic Web (pp. 142–155). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/978-3-540-76298-0_11.
(Gause & Weinberg, 1989)	Gause, D. Weinberg, G. (1989). *Exploring Requirements Quality Before Design*, Dorset House, 1989.
(Gemino & Wand, 2004)	Gemino, A. & Wand, Y. (2004). *A framework for empirical evaluation of conceptual modeling techniques*. Requirements Engineering, 9(4), 248–260. http://doi.org/10.1007/s00766-004-0204-6.
(Gilvaz & Leite, 1995)	Gilvaz A. P. Leite, J. C. S.P. (1995) *FAES: A Case Tool for Information Acquisition*. Case .95, Proceedings of the Seventh Intl. Work on Computer Aided Software engineering. IEEE Computer Society Press, pp. 260-269.
(Glinz, 2007)	Glinz, M. (2007). *On Non-Functional Requirements*. In *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International* (pp. 21–26). doi:10.1109/RE.2007.45.

(Glinz & Wieringa, 2007)	Glinz, M. & Wieringa, R. J. (2007). *Guest Editors' Introduction:* Stakeholders in Requirements Engineering. IEEE Software, 24(2), 18–20.
(Goguen & Linde, 1993)	Goguen, J. A. & Linde, C. (1993). *Techniques for requirements elicitation*. In Requirements Engineering, 1993. Proceedings of IEEE International Symposium on (pp. 152–164). Available online at http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=324822.
(Gordijn, J. & Akkermans, 2003)	Gordijn, J. & Akkermans, J. M. (2003). *Value-based requirements engineering: exploring innovative e-commerce ideas*. Requirements Engineering, 8(2), 114–134. doi:10.1007/s00766-003-0169-x.
(Gottesdeiner, 2002)	Gottesdeiner, E. (2002). *Requirements by Collaboration*, Addison-Wesley, 2002.
(Greenspan, Mylopoulos & Borgida, 1994)	Greenspan, S. Mylopoulos, J. & Borgida, A. (1994). *On formal requirements modeling languages: RML revisited*. In Proceedings of the 16th international conference on Software engineering (pp. 135–147). IEEE Computer Society Press. Available online at http://dl.acm.org/citation.cfm?id=257754.
(Groza, Schutz & Handschuh, 2007)	Groza, T. Schutz, A. & Handschuh, S. (2007). *SALT: a semantic approach for generating document representations*. In Proceedings of the 2007 ACM symposium on Document engineering (pp. 171–173). ACM. Available online at http://dl.acm.org/citation.cfm?id=1284462.
(Gruber, 1993)	Gruber, T. R. (1993). *A translation approach to portable ontology specifications. Knowledge Acquisition, 5*(2), 199–220.
(Guarino, 1998)	Guarino, N. (1998). *Formal Ontology and Information Systems* (pp. 3–15). Presented at the Proceedings of FOIS'98, Trento, Italy: Amsterdam, IOS Press.
(Hadar, Soffer & Kenzi, 2014)	Hadar, I. Soffer, P. & Kenzi, K. (2014). *The role of domain knowledge in requirements elicitation via interviews: an exploratory study*. Requirements Engineering, 19(2), 143–159.
(Hassan & Salim, 2004)	Hassan, S. & Salim, S. S. (2004). *A Tool to Support Collaborative Requirements Elicitation using Viewpoint Approach*. In 1st International Conference on Informatics, Izmir, Turkey, September (pp. 1–4).
(Haumer, Pohl & Weidenhaupt, 1998)	Haumer, P. Pohl, K. & Weidenhaupt, K. (1998). *Requirements elicitation and validation with real world scenes*. Software Engineering, IEEE Transactions on, 24(12), 1036–1054. doi:10.1109/32.738338.

(Havey, 2005)	Havey, M. (2005). C115. O'Reilly Media, Inc. Available online at http://books.google.com/books?hl=en&lr=&id=RFBHCaMtlfI C&oi=fnd&pg=PP8&dq=essential+business+process+modelli ng+havey&ots=wX3LQ0l1H1&sig=A6GPCSsuGHi-PiD5BYsHuWa-h7w.
(Hepp, Leymann, Domingue, Wahler & Fensel 2005)	Hepp, M. Leymann, F. Domingue, J. Wahler, A. & Fensel, D. (2005). *Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management*. IEEE International Conference on e-Business Engineering (ICEBE 2005), 18th -21st October, Beijing, China. IEEE Computer Society, ISBN 0-7695-2430-3. pp. 535-540.
(Hepp, Roman, Hepp & Roman, 2007)	Hepp, M. Roman, D. M. Hepp, D. Roman (2007). *An Ontology Framework for Semantic Business Process Management*. Proceedings of the 8th international conference Wirtschaftsinformatik, February 28 - March 2, 2007, Karlsruhe. In: Oberweis, A; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.; Schmizler, B.: eOrganisation: Service-, Prozess, Market-Engineering, Vol. 1, Universitaetsverlag Karlsruhe, pp. 423-440. Available online at http://www.heppnetz.de/files/hepp-roman-an-ontology-framework-for-SBPM-WI2007.pdf
(Herbsleb & Moitra, 2001)	Herbsleb, J. D. & Moitra, D. (2001). *Global software development*. Software, IEEE, 18(2), 16–20.
(Hickey & Davis, 2003)	Hickey, A. M. & Davis, A. M. (2003). *Requirements elicitation and elicitation technique selection: a model for two knowledge-intensive software development processes*. In Proceedings of the 36th Annual Hawaii International Conf. on System Sciences (pp. 6–9).
(Higher Education Authority, 2014)	Higher Education Authority. (2014). *STATISTICS*. Available online at http://www.hea.ie/en/statistics/overview.
(Hoffman, Shadbolt, Burton & Klein, 1995)	Hoffman, R. R. Shadbolt, N. R. Burton, A. M. & Klein, G. (1995). *Eliciting knowledge from experts: A methodological analysis*. Organizational Behavior and Human Decision Processes, 62(2), 129–158.
(Holbrook III, 1990)	Holbrook III, H. (1990). *A scenario-based methodology for conducting requirements elicitation. ACM SIGSOFT Software Engineering Notes*, *15*(1), 95–104.
(RE'15, n.d.)	RE'15. (n.d.). Available online at http://re15.org/.
(Hsia, Samuel, Gao, Kung, Toyoshima & Chen, 1994)	Hsia, P. Samuel, J. Gao, J. Kung, D. Toyoshima, Y. & Chen, C. (1994). *Formal approach to scenario analysis*. IEEE Software, 11(2), 33–41.

(Hull, Jackson & Dick, 2010)	Hull, E. Jackson, K. & Dick, J. (2010). *Requirements Engineering* (3rd edition). London ; New York: Springer.
(IBM - SPSS software - Ireland, 2015)	IBM - SPSS software - Ireland. (2015). Available online at http://www-01.ibm.com/software/ie/analytics/spss/.
(Institute of Electrical and Electronic Engineers, 1998)	Institute of Electrical and Electronic Engineers. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Standard 830-1998, Institute of Electrical and Electronic Engineers, New York, 1998.
(International Institute of business analysis, 2009)	International Institute of Business Analysis. (2009). *A guide to the business analysis body of knowledge (Babok Guide) Version 2.0*. International Institute of Business Analysis.
(International Requirements Engineering Board, n.d.)	International Requirements Engineering Board. (n.d.). Available online at http://www.ireb.org/en/home.html.
(Java Platform v7.5, n.d.)	Java Platform v7.5. (n.d.). Available online at http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u5-oth-JPR.
(JDOM v1.1., 2007)	JDOM v1.1. (2007). Available online at http://www.jdom.org/dist/binary/archive/.
(Jiang, 2005)	Jiang L (2005), *A framework for requirements engineering process development*. University of Calgary, PhD Thesis, Sept. 2005.
(Johnson & Lederer, 1991)	Johnson, A.M. & Lederer, A.L. (1991). *The effect of communication frequency and channel richness on the convergence between chief executive and chief information officers*. Journal of Management Information Systems / Fall 2005, Vol. 22, No. 2, pp. 227-252.
Johnson & Lederer, 2005)	Johnson, A.M. & Lederer, A.L. (2005). *The effect of communication frequency and channel richness on the convergence between chief executive and chief information officers*. Journal of Management Information Systems / Fall 2005, Vol. 22, No. 2, pp. 227-252.
(Kaljurand, 2013)	Kaljurand, K. (2013). *APE (ACE Parser) v6.0*. February 26, 2015, Available online at http://attempto.ifi.uzh.ch/site/resources/.
(Kaljurand & Fuchs, 2007)	Kaljurand, K. & Fuchs, N. E. (2007). *Verbalizing OWL in Attempto Controlled English*. In OWLED (Vol. 258). Available online at http://owl1-1.googlecode.com/svn-history/r599/trunk/www.webont.org/owled/2007/PapersPDF/submission_22.pdf.

(Kaljurand & Fuchs, 2006)	Kaljurand, K. and Fuchs, N. E. (2006). *Bidirectional mapping between OWL DL and Attempto Controlled English*. In Fourth Workshop on Principles and Practice of Semantic Web Reasoning, Budva, Montenegro.
(Kazman & Bass, 1994)	Kazman, R. & Bass, L. (1994). *Toward deriving software architectures from quality attributes*. DTIC Document. Available online at http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA283827.
(Kellner, Madachy & Raffo, 1999)	Kellner, M.I. Madachy, R.J. and Raffo, D.M. (1999). *Software Process Modeling and Simulation: Why, What, How*. Journal of Systems and Software, Vol. 46, No. 2/3.
(Kensing, Simonsen & Bødker, 1998)	Kensing F, Simonsen J, Bødker K. (1998). *MUST: A method for participatory design*. Human-Computer Interaction 1998; 13 (2): 167-198.
(Kimmond, 1995)	Kimmond, R. M. (1995). *Survey into the acceptance of prototyping in software development*. In *Sixth IEEE International Workshop on Rapid System Prototyping, 1995. Proceedings* (pp. 147–152). doi:10.1109/IWRSP.1995.518584.
(Kolfschoten & De Vreede, 2007)	Kolfschoten, G. L. & De Vreede, G.-J. (2007). *The collaboration engineering approach for designing collaboration processes. In Groupware: design, implementation, and use* (pp. 95–110). Springer. Available online at http://link.springer.com/chapter/10.1007/978-3-540-74812-0_8.
(Kolfschoten, Appelman, Briggs & de Vreede, 2004)	Kolfschoten, G. L. Appelman, J. H. Briggs, R. O. & de Vreede, G. (2004). *Recurring patterns of facilitation interventions in GSS sessions*. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004 (p. 10 pp.–). doi:10.1109/HICSS.2004.1265074.
(Kotonya & Sommerville, 1992)	Kotonya, G. & Sommerville, I. (1992). *Viewpoints for requirements definition*. Software Engineering Journal, 7(6), 375–387.
(Kotonya & Sommerville, 1996)	Kotonya, G. & Sommerville, I. (1996). *Requirements Engineering With Viewpoints*. Software Engineering Journal, 11, 5–18.
(Kruchten, Capilla & Duñeas, 2009)	Kruchten, P. Capilla, R. & Duñeas, J. C. (2009). *The Decision View's Role in Software Architecture Practice*. IEEE Software, 26(2), 36–42. doi:10.1109/MS.2009.52.
(Kuhn, 2008)	Kuhn, T. (2008). *Acewiki: A natural and expressive semantic wiki*. Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges, CEUR Workshop Proceedings. Available online at http://arxiv.org/abs/0807.4618.

(Kuhn, 2008a)   Kuhn, T. (2008a). *Combining Semantic Wikis and Controlled Natural Language.* Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), CEUR Workshop Proceedings, 2008.

(Kuhn, 2008b)   Kuhn, T. (2008b). *Acewiki: Collaborative ontology management in controlled natural language.* In Proceedings of the 3rd Semantic Wiki Workshop, CEUR Workshop Proceedings. Available online at http://arxiv.org/abs/0807.4623.

(Kyng & Greenbaum, 1991)   Kyng, M. & Greenbaum, J. (1991). *Design at Work: Cooperative design of computer systems.* Lawrence Erlbaum Associates, Incorporated. Available online at http://forskningsbasen.deff.dk/Share.external?sp=Sda7e9f30-ed67-11df-a891-000ea68e967b&sp=Sau.

(Jiang, Eberlein & Far, 2008)   Jiang, L. Eberlein, A. and Far, B. (2008). *A case study validation of a knowledge-based approach for the selection of requirements engineering techniques.* Requirements Engineering, vol. 13, no. 2, pp. 117–146.

(Lam, 2006)   Lam, G.S.W. (2006). *Business Rules vs. Business Requirements.* Business Rules Journal, Vol. 7, No. 5 (May 2006),
URL: http://www.BRCommunity.com/a2006/b290.html.

(Laudon & Laudon, 2006)   Laudon, K. Laudon, J. (2006), *Management Information Systems: managing the digital firm.* 9th edition, Pearson Prentice Hall, upper saddle river, New Jersey.

(Li, Jin, Xu & Lu, 2011)   Li, G. Jin, Z. Xu, Y. & Lu, Y. (2011). *An Engineerable Ontology Based Approach for Requirements Elicitation in Process Centered Problem Domain.* In H. Xiong & W. B. Lee (Eds.), Knowledge Science, Engineering and Management (pp. 208–220). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/978-3-642-25975-3_19.

(Lin, Fox & Bilgic, 1996)   Lin, J. Fox, M. S. & Bilgic, T. (1996). *A Requirement Ontology for Engineering Design.* Concurrent Engineering, 4(3), 279–291. doi:10.1177/1063293X9600400307.

(Lind & Zmud, 1991)   Lind, M. R. & Zmud, R. W. (1991). *The Influence of a Convergence in Understanding between Technology Providers and Users on Information Technology Innovativeness.* Organization Science, 2(2), 195-217.

(List & Korherr, 2005)	List, B. and Korherr, B. (2005). *A UML 2 Profile for Business Process Modelling*. In Proceedings of the 1st International Workshop on Best Practices of UML (BP-UML 2005) at the 24th International Conference on Conceptual Modeling (ER 2005), Springer Lecture Notes in Computer Science LNCS 3770, 2005.
(List & Korherr, 2006)	List, B. Korherr, B. (2006). *An Evaluation of Conceptual Business Process Modelling Languages*. IN: proceedings of SAC'06, April 23-27, 2006, Dijon, France. ACM pp 1532-1539.
(Loucopoulos & Champion, 1988)	Loucopoulos, P. & Champion, R. (1988). *Knowledge-based approach to requirements engineering using method and domain knowledge*. Knowledge-Based Systems, 1(3), 179–187. doi:10.1016/0950-7051(88)90076-7.
(Luftman, 2000)	Luftman, J. (2000). *Assessing Business-IT Alignment Maturity*. Communications of the Association for Information Systems, 4(1). Available online at http://aisel.aisnet.org/cais/vol4/iss1/14.
(Luftman & Ben-Zvi, 2010a)	Luftman, J. & Ben-Zvi, T. (2010a). *Key Issues for IT Executives 2009: Difficult Economy's Impact on IT*. MIS Quarterly Executive, 9(1). Available online at http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=15401960&AN=48637857&h=otHTMVn7iD7LubInqrja7LkSwxf2pyMMXKqXLGxCpokyxsfHlG9POuk%2FmJgYXuNjKR1FG2BdzNuV6EeN1VI%2Bsg%3D%3D&crl=c.
(Luftman & Ben-Zvi, 2010b)	Luftman, J. & Ben-Zvi, T. (2010b). *Key Issues for IT Executives 2010: Judicious IT Investments Continue Post-Recession*. MIS Quarterly Executive, 9(4). Available online at http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=15401960&AN=58657255&h=TaA0oEyo%2F8qgh73ySTrWQZAa463mxJXltPCAiQx3NlKqqihvleP3LaeXfZ3S98QY3e4qhnb6LgQ%2BueMJW8Nv1g%3D%3D&crl=c.
(Luftman & Ben-Zvi, 2011)	Luftman, J. & Ben-Zvi, T. (2011). *Key Issues for IT Executives 2011: Cautious Optimism in Uncertain Economic Times*. MIS Quarterly Executive, 10(4). Available online at http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=15401960&AN=67793224&h=ijyReAli4Zcy%2FUP324SI9lP%2BTBhbtDyUA5GVxKwAccAsxmcOlcteUeZCVyf5P7E1ZC47CDDVBagFVfp02%2BIVCQ%3D%3D&crl=c.

(Luftman & Derksen, 2012)	Luftman, J. & Derksen, B. (2012). *Key Issues for IT Executives 2012: Doing More with Less*. MIS Quarterly Executive, 11(4). Available online at http://search.ebscohost.com/login.aspx?direct=true&profile=e host&scope=site&authtype=crawler&jrnl=15401960&AN=83 863564&h=poDHfk5vpQN8F4Mi9wUuMIhjwUFSrYdm0%2 BTPpSFVcFizAKB0uTRECUydVozTMAejk4gU8BTCl6Y0 LeAMqutGKw%3D%3D&crl=c.
(Luftman, Kempaiah & Nash, 2006)	Luftman, J. Kempaiah, R. & Nash, E. (2006). *Key issues for IT executives 2005*. MIS Quarterly Executive, 5(2). Available online at http://search.ebscohost.com/login.aspx?direct=true&profile=e host&scope=site&authtype=crawler&jrnl=15401960&AN=22 198563&h=ATzkcKK%2BZY%2Fkft3VhV6GvGVxaTTlDK D7pKYuTZCkovu1HnQoNORsu%2FxtO2lOgffjnOZdkGd2S W%2Bhkge0pXgtmQ%3D%3D&crl=c.
(Luftmann & Kempaiah, 2008)	Luftmann, J. & Kempaiah, R. (2008). *Key Issues for IT Executives 2007. MIS Quarterly Executive*, 7(2). Available online at http://search.ebscohost.com/login.aspx?direct=true&profile=e host&scope=site&authtype=crawler&jrnl=15401960&AN=32 927051&h=RWYlcWVL2GjGQwsJBBVKaICwQM8ewM16 k5PnhR4MmwMenoETn4wKhExZu61rGaEuPHAvQBRVm RMEVk4F2KeGqw%3D%3D&crl=c.
(Macaulay, 1996)	Macaulay LA. (1996). *Requirements Engineering*. Springer-Verlag, London, 1996.
(Macaulay, 1995)	Macaulay, L. (1995). *Cooperation in understanding user needs and requirements*. Computer Integrated Manufacturing Systems, *8*(2), 155–165.
(Machado, Ramos & Fernandes, 2005)	Machado, R. J. Ramos, I. & Fernandes, J. M. (2005). *Specification of requirements models*. In Engineering and managing software requirements (pp. 47–68). Springer. Available online at http://link.springer.com/chapter/10.1007/3-540-28244-0_3.
(Maciaszek, 2001)	Maciaszek, L. (2001). *Requirements Analysis and System Design*, Addison-Wesley, 2001.
(Maiden, 2006)	Maiden, N. (2006). *Servicing Your Requirements. IEEE Software*, *23*(5), 14–16. doi:10.1109/MS.2006.140.
(Maiden & Rugg 1996)	Maiden, N. A. M. & Rugg, G. (1996). *ACRE: selecting methods for requirements acquisition*. Software Engineering Journal, 11(3), 183–192.
(Martin, 1991)	Martin, J. (1991). *Rapid application development*. Macmillan Publishing Co. Inc. Available online at http://dl.acm.org/citation.cfm?id=103275.

(Mishra & Mohanty, 2011)	Mishra, J. & Mohanty, A. (2011). *Software Engineering* (1 edition). Pearson.
(Mitchell & Jolley, 2004)	Mitchell, M. & Jolley, J. (2004). *Research Design Explained* (5th Edition.). Belmont, CA: Wadsworth/Thomson Learning. Available online at http://www.abebooks.com/Research-Design-Explained-5th-Mitchell-Mark/930507386/bd.
(Monsalve, April & Abran, 2012)	Monsalve, C. April, A. & Abran, A. (2012). *On the expressiveness of business process modeling notations for software requirements elicitation*. In IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society (pp. 3132–3137). IEEE. Available online at http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6389398.
(Morse, 2000)	Morse, J.M. (2000). *Determining sample size*. Qualitative Health Research (10)(1), 3–5.
(Muller, Haslwanter & Dayton, 1997)	Muller, M. J. Haslwanter, J. H. & Dayton, T. (1997). Participatory practices in the software lifecycle. Handbook of Human-Computer Interaction, 2, 255–297.
(Mullery, 1979)	Mullery, G. (1979). *CORE - A Method for Controlled Requirements Specification*. in *4th Int. Conf. on Software Engineering*. Munich. IEEE Press.
(Musen, 1992)	Musen, M. A. (1992). *Dimensions of knowledge sharing and reuse*. Computers and Biomedical Research, *25*(5), 435–467.
(mxGraph v1.10.0.2., 2012)	mxGraph v1.10.0.2. (2012). Available online at https://www.jgraph.com/.
(Mylopoulos, Borgida, Jarke & Koubarakis, 1990)	Mylopoulos, J. Borgida, A. Jarke, M. & Koubarakis, M. (1990). *Telos: Representing Knowledge About Information Systems*. ACM Trans. Inf. Syst. 8(4), 325–362. doi:10.1145/102675.102676.
(Mylopoulos, Chung & Nixon 1992)	Mylopoulos, J. Chung, L. & Nixon, B. (1992). *Representing and using nonfunctional requirements: a process-oriented approach*. IEEE Transactions on Software Engineering, 18(6), 483–497. doi:10.1109/32.142871.
(Nanz, 2010)	Nanz, S. (2010). *The Future of Software Engineering*. Springer Science & Business Media.
(Neches, Fikes, Finin, Gruber, Patil, Senator & Swartout, 1991)	Neches, R. Fikes, R. E. Finin, T. Gruber, T. Patil, R. Senator, T. & Swartout, W. R. (1991). *Enabling technology for knowledge sharing*. AI Magazine, 12(3), 36.
(NetBeans IDE v6.8, n.d.)	NetBeans IDE v6.8. (n.d.). February 26, 2015, Available online at https://netbeans.org/downloads/6.8/.

(Nguyen, Vo, Lumpe & Grundy, 2014)	Nguyen, T. H. Vo, B. Q. Lumpe, M. & Grundy, J. (2014). *KBRE: A Framework for Knowledge-based Requirements Engineering*. Software Quality Control, 22(1), 87–119. http://doi.org/10.1007/s11219-013-9202-6.
(Nielsen, 1993)	Nielsen, J. (1993). *Usability Engineering*. Academic Press. Chapter 5, p. 115. http://hcibib.org/perlman/question.cgi?form=NHE.
(Nuseibeh & Easterbrook, 2000)	Nuseibeh, B. & Easterbrook, S. (2000). *Requirements engineering: a roadmap*. In Proceedings of the Conference on the Future of Software Engineering (pp. 35–46).
(Nuseibeh, Kramer & Finkelstein, 1994)	Nuseibeh, B. Kramer, J. & Finkelstein, A. (1994). *A framework for expressing the relationships between multiple views in requirements specification*. Software Engineering, IEEE Transactions on, 20(10), 760–773.
(O'Loughlin, 2010)	O'Loughlin, E. (2010). *An Introduction to Business Systems Analysis: Problem Solving Techniques and Strategies*. Dublin, Ireland: The Liffey Press.
(Object Management Group, 2004)	Object Management Group. (2004). *Business Motivation Model*. October 31, 2014, Available online at http://www.omg.org/spec/BMM/1.2/PDF/.
(Object Management Group, 2011)	Object Management Group. (2011). *Business Process Model and Notation (BPMN)* Version 2.0. Available online at http://www.omg.org/spec/BPMN/2.0/PDF/.
(OWL 2 Web Ontology Language Document Overview (Second Edition), 2012)	OWL 2 Web Ontology Language Document Overview (Second Edition). (2012). Available online at http://www.w3.org/TR/owl2-overview/.
(Package javax.xml.transform, n.d.)	Package javax.xml.transform. (n.d.). Available online at http://docs.oracle.com/javase/7/docs/api/javax/xml/transform/package-summary.html on 4/7/2014.
(Pellet: OWL 2 Reasoner v2.3.0., 2011)	Pellet: OWL 2 Reasoner v2.3.0. (2011). Available online at http://clarkparsia.com/pellet/.
(Pohl & Rupp, 2011)	Pohl, K. & Rupp, C. (2011). *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant* (1 edition). Santa Barbara, CA : Sebastopol, CA: Rocky Nook.
(Pool, 2006)	Pool, J. (2006). *Can Controlled Languages Scale to the Web?* In International Workshop on Controlled Language Applications 5. Available online at http://www.mt-archive.info/CLAW-2006-Pool.pdf.

(Potts, Takahashi & Antón, 1994)	Potts, C. Takahashi, K. & Antón, A. I. (1994). *Inquiry-based requirements analysis*. IEEE Software, 11(2), 21–32.
(Pratt-Hartmann, 2003)	Pratt-Hartmann, I. (2003). *A Two-Variable Fragment of English*. Journal of Logic, Language and Information, *12*(1), 13–45.
(Pressman, 2010)	Pressman, R. (2010). *Software Engineering: A practioners Approach*. Seventh Edition, McGraw and Hill, New York, NY.
(Pressman & Maxim, 2014)	Pressman, R. S. & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8th edition). New York, NY: McGraw-Hill Higher Education.
(Preston, Karahanna & Rowe, 2006)	Preston, D. S. Karahanna, E. & Rowe, F. (2006). *Development of Shared Understanding Between the Chief Information Officer and Top Management Team in U.S. and French Organizations: A Cross-Cultural Comparison*. IEEE Transactions on Engineering Management, 53(2), 191-206. doi: 10.1109/TEM.2006.872244.
(Protégé v4.2., 2013)	Protégé v4.2. (2013). Available online at http://protege.stanford.edu/.
(Qualitative Research | Data Analysis Software | NVivo, n.d.)	Qualitative Research	Data Analysis Software	NVivo. (n.d.). Available online at http://www.qsrinternational.com/.
(Quality and Qualifications Ireland, 2012)	Quality and Qualifications Ireland. (2012). *National Framework of Qualifications (NFQ)*. March 9, 2015, Available online at http://www.qqi.ie/Pages/National-Framework-of-Qualifications-(NFQ).aspx.
(Yeh, 1982)	Yeh, R. T. (1982). *Requirements Analysis-A Management Perspective*, Proc. COMPSAC '82, pp. 410-416.
(RDF - Semantic Web Standards, 2014)	RDF - Semantic Web Standards. (2014). Available online at http://www.w3.org/RDF/.
(Reich & Benbasat, 2000)	Reich, B. H. & Benbasat, I. (2000). *Factors that influence the social dimension of alignment between business and information technology objectives*. MIS Quarterly, 81–113.
(Robertson & Robertson, 2012)	Robertson, S. & Robertson, J. (2012). *Mastering the Requirements Process: Getting Requirements Right* (3 edition.). Upper Saddle River, NJ: Addison-Wesley Professional.
(Robertson & Robertson, 1999)	Robertson, S. and Robertson, J. (1999). *Mastering the Requirements Process*. Addison-Wesley, 1999.
(Rogers & Kincaid,	Rogers, E.M. and Kincaid, D.L. (1981). *Communication*

1981)	*Networks*. New York: Free Press, 1981.
(Roman, 1985)	Roman, G. (1985). *A taxonomy of current issues in requirements engineering. Computer*, *18*(4), 14–23. doi:10.1109/MC.1985.1662861.
(Rospocher & Serafini, 2009)	Rospocher, M. & Serafini, L. (2009). *BPMN Ontology V1.1 | DKM*. February 26, 2015, Available online at https://dkm.fbk.eu/bpmn-ontology.
(Rospocher, Ghidini & Serafini, 2009)	Rospocher, M. Ghidini, C. & Serafini, L. (2009). *BPMN Ontology | DKM*. February 24, 2015, Available online at https://dkm.fbk.eu/bpmn-ontology.
(Rospocher, Ghidini & Serafini, 2014)	Rospocher, M. Ghidini, C. Serafini, L. (2014). *An ontology for the Business Process Modelling Notation*. Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS2014, September, 22-25, 2014, Rio de Janeiro, Brazil, vol. 267, pp. 133-146, IOS Press, 2014.
(Saiedian & Dale, 2000)	Saiedian, H. & Dale, R. (2000). *Requirements engineering: making the connection between the software developer and customer*. Information and Software Technology, 42(6), 419–428.
(Sawyer, Sommerville & Viller, 1996)	Sawyer, P. Sommerville, I. & Viller, S. (1996). *PREview: tackling the real concerns of requirements engineering*. Cooperative Systems Engineering Group, Computing Department, Lancaster University, Lancaster, Technical Reports No: CSEG/5/96.
(Saxonica, 2007)	Saxonica, M. K. (2007). *XSL Transformations (XSLT)* Version 2.0. February 26, 2015, Available online at http://www.w3.org/TR/xslt20/.
(Scacchi, 2004)	Scacchi, W. (2004). *Socio-technical design*. The Encyclopedia of Human-Computer Interaction, *1*, 656–659.
(Schuler & Namioka, 1993)	Schuler, D. & Namioka, A. (1993). Participatory Design: Principles and Practices. Hillsdale, N.J: CRC / Lawrence Erlbaum Associates.
(Schwitter & Tilbrook, 2004)	Schwitter, R. & Tilbrook, M. (2004). *Controlled natural language meets the semantic web*. In Proceedings of the Australasian Language Technology Workshop (Vol. 2, pp. 55–62). Citeseer.
(Schwitter, 2005)	Schwitter, R. (2005). *A controlled natural language layer for the semantic web*. IN: Proceedings of the 25th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI 2005), 12-14 December 2005, Cambridge, UK.Springer, Berlin. pp. 425-434.

(Selby, 2007)	Selby, R. W. (2007). *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research*. John Wiley & Sons.
(Shaw & Woodward, 1990)	Shaw, M. L. G. & Woodward, J. B. (1990). *Modeling expert knowledge*. Knowledge Acquisition, 2(3), 179–206. doi:10.1016/S1042-8143(05)80015-9.
(Shibaoka, Kaiya & Saeki, 2007)	Shibaoka, M. Kaiya, H. & Saeki, M. (2007). *GOORE : Goal-Oriented and Ontology Driven Requirements Elicitation* Method. In J.-L. Hainaut, E. A. Rundensteiner, M. Kirchberg, M. Bertolotto, M. Brochhausen, Y.-P. P. Chen, E. Zimányie (Eds.), Advances in Conceptual Modeling – Foundations and Applications (pp. 225–234). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/978-3-540-76292-8_28.
(Siegemund, Thomas, Zhao, Pan & Assmann, 2011)	Siegemund, K. Thomas, E. J. Zhao, Y. Pan, J. & Assmann, U. (2011). *Towards ontology-driven requirements engineering*. In Workshop Semantic Web Enabled Software Engineering at 10th International Semantic Web Conference (ISWC), Bonn. Available online at http://www.researchgate.net/profile/Katja_Siegemund/publication/236592768_SWESE2011_Ontology-drivenRE/links/004635182664a03f16000000.pdf.
(Smith, 1776)	Smith, A. (1776). *An Inquiry into the Nature and Causes of the Wealth of Nations*. Edwin Cannan's Annotated Edition. Available online at http://www.wwww4.com/read_book/an_inquiry_into_the_nature_and_causes_of_the_wealth_of_nations_1589586.pdf.
(Software Engineering Institute Requirements Engineering Project, 1991)	Software Engineering Institute Requirements Engineering Project. (1991). *Requirements Engineering and Analysis* Workshop Proceedings. Technical Report CMU/SEI-91-TR-30 or ESD-TR-91-30, Software Engineering Institute, Pittsburgh, PA.
(Sommerville, 2001)	Sommerville I. (2001). *Software Engineering*, 6h Edition, Addison-Wesley, Harlow, England.
(Sommerville, 2005)	Sommerville, I. (2005). *Integrated requirements engineering: a tutorial. IEEE Software*, *22*(1), 16–23. doi:10.1109/MS.2005.13.
(Sommerville, 2010)	Sommerville, I. (2010). *Software Engineering: International Version* (9 edition.). Boston: Pearson.

(Sommerville & Kotonya, 1998)	Sommerville, I. & Kotonya, G. (1998). *Requirements engineering: processes and techniques*. John Wiley & Sons, Inc. Available online at http://dl.acm.org/citation.cfm?id=552009.
(Sommerville, Sawyer & Viller, 1998)	Sommerville, I. et Sawyer, P. & Viller, S. (1998). *Viewpoints for requirements elicitation: a practical approach*. In Proc. Third International Conference on Requirements Engineering, IEEE Computer Society Press, 1998, pp. 74-81.
(Stynes, Conlan & O'Sullivan, 2008)	Stynes, P. Conlan, O. O'Sullivan, D. (2008). *Towards a Simulation-based Communication Tool to Support Semantic Business Process Management*. IN: Proceedings of the Fourth International Workshop on Semantic Business Process Management in Proceedings of Workshops held at the Fifth European Semantic Web Conference, (ESWC08), 2nd June, Tenerife, Canary Islands, Spain.
(Stynes, Conlan & O'Sullivan, 2009)	Stynes, P. Conlan, O. O'Sullivan, D. (2009). *Simulation-based Communication Tool: an Enabler for Collaborative Decision Making*. IN: Proceedings of the Workshop on Controlled Natural Language (CNL) 2009, 8th-10th June, Marettimo Island, Italy.
(Sutcliffe, 2003)	Sutcliffe, A. (2003). *Scenario-based requirements engineering*. In Requirements Engineering Conference, *2003. Proceedings. 11th IEEE International* (pp. 320–329). doi:10.1109/ICRE.2003.1232776.
(Sutcliffe & Ryan, 1998)	Sutcliffe, A. Ryan, M. (1998), *Experience with SCRAM, a Scenario Requirements Analysis Method*. Third International Conference on Requirements Engineering, IEEE Computer Society Press, 1998, pp. 164-171.
(Andrews, Curbera, Dholakia, Goland, Klein, Leymann, Liu, Roller, Smith, Thatte, Trickovic & Weerawarana, 2003)	Andrews, T. Curbera, F. Dholakia, H. Goland, Y. Klein, J. Leymann, F. Liu, K. Roller, D. Smith, D. Thatte, S. Trickovic, I. Weerawarana, S.(2003) *Business Process Execution Language for Web Services*, Version 1.1. Available online at http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf [Accessed on 22/08/2009]
(Takeda, Shiomi, Kawai & Ohiwa, 1993)	Takeda, N. Shiomi, A. Kawai, K. & Ohiwa, H. (1993). *Requirement analysis by the KJ editor*. Requirements Engineering, 1993. Proceedings of IEEE International Symposium on, 98–101. doi:10.1109/ISRE.1993.324829.
(TeamWave Software Ltd, 2001)	TeamWave Software Ltd. (2001). Available at URL http://www.teamwave.com/.
(TrAX API, n.d.)	TrAX API. (n.d.). Available at http://xalan.apache.org/old/xalan-j/trax.html.

(Using the JAXP Transform APIs, n.d.)	Using the JAXP Transform APIs. (n.d.). Available online at http://xalan.apache.org/old/xalan-j/trax.html.
(Van Heesch & Avgeriou, 2011)	Van Heesch, U. & Avgeriou, P. (2011). *Mature Architecting - A Survey about the Reasoning Process of Professional Architects*. In 2011 9th Working IEEE/IFIP Conference on Software Architecture (WICSA) (pp. 260–269). doi:10.1109/WICSA.2011.42.
(Van Lamsweerde, 2009)	Van Lamsweerde, A. (2009). *Requirements Engineering from system goals to UML models to Software Specifications*. John Wiley & Sons, Chichester, England.
(Veres, Sampson, Bleistein, Cox & Verner, 2009)	Veres, C. Sampson, J. Bleistein, S. J. Cox, K. & Verner, J. (2009). *Using Semantic Technologies to Enhance a Requirements Engineering Approach for Alignment of IT with Business Strategy*. In International Conference on Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09 (pp. 469–474). doi:10.1109/CISIS.2009.199.
(Vernadat, 1997)	Vernadat, F. B. (1997). Enterprise Modelling Languages. In K. Kosanke & J. G. Nell (Eds.). *Enterprise Engineering and Integration* (pp. 212–224). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/978-3-642-60889-6_24.
(Vliet, 2008)	Vliet, H. van. (2008). *Software Engineering: Principles and Practice* (3rd Edition edition). Chichester, England ; Hoboken, NJ: John Wiley & Sons.
(W3C Document Object Model, n.d.)	W3C Document Object Model. (n.d.). Available online at http://www.w3.org/DOM/.
(W3C XML Schema v2.0., 2004)	W3C XML Schema v2.0. (2004). Available online at http://www.w3.org/XML/Schema.
(Walia & Carver, 2009)	Walia, G. S. & Carver, J. C. (2009). *A systematic literature review to identify and classify software requirement errors*. Information and Software Technology, 51(7), 1087–1109. doi:10.1016/j.infsof.2009.01.004.
(Wang, Xiong, Zhou & Yu, 2007)	Wang, C. Xiong, M. Zhou, Q. & Yu. Y. (2007). *Panto: a portable natural language interface to ontologies*. IN: Proceedings of the 4th European Semantic Web Conference (ESWC 2007), 3-7 June, Innsbruck, Austria. Berlin, Springer-Verlagg. pp. 473-487.
(Web Services Activity Statement, n.d.)	Web Services Activity Statement. (n.d.). Available online at http://www.w3.org/2002/ws/Activity.

(Weidenhaupt, Pohl, Jarke & Haumer, 1998)	Weidenhaupt, K. Pohl, K. Jarke, M. & Haumer, P. (1998). *Scenarios in system development: current practice*. IEEE Software, 15(2), 34–45. doi:10.1109/52.663783.
(White, 2004)	White, S. (2004). *Business Process Modeling Notation (BPMN)*, Version 1.0. Available online at http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf [Accessed on 22nd August, 2009].
(Wiegers & Beatty, 2013)	Wiegers, K. E. & Beatty, J. (2013). *Software Requirements 3* (3 edition). Redmond, Washington: Microsoft Press.
(Wohed, Van der Aalst, Dumas & Hofstede, 2003)	Wohed, P. Van der Aalst, W.M.P. Dumas, M. Ter Hofstede, A.H.M. (2003). A*nalysis of Web Services Composition Languages: The Case of BPEL4WS*. IN: Proceedings of the 22nd International Conference on Conceptual Modeling, (ER 2003), Chicago, IL, USA, October 13-16. I.-Y. Song et al.. (Eds.): ER 2003, LNCS 2813, pp. 200–215.
(Wood & Silver, 1989)	Wood J, Silver D. (1989), *Joint application design: how to design quality systems in 40% less time*. Wiley, NY, 1989.
(Wood & Silver, 1995)	Wood, J. & Silver, D. (1995). *Joint application development*. John Wiley & Sons, Inc. Available online at http://dl.acm.org/citation.cfm?id=200622.
(Xalan-Java Version 2.7.1., 2007)	Xalan-Java Version 2.7.1. (2007). Available online at http://xalan.apache.org/old/xalan-j/#current.
(XML Path Language (XPath) Version 1.0., 1999)	XML Path Language (XPath) Version 1.0. (1999). Available online at http://www.w3.org/TR/xpath/.
(XQuery 1.0: An XML Query Language (Second Edition), 2010)	XQuery 1.0: An XML Query Language (Second Edition). (2010). Available online at http://www.w3.org/TR/xquery/.
(Yang, Bhuta, Boehm & Port, 2005)	Yang, Y. Bhuta, J. Boehm, B. & Port, D. N. (2005). *Value-based processes for COTS-based applications*. Software, IEEE, 22(4), 54–62.
(Yildiz & Miksch, 2007)	Yildiz, B. & Miksch, S. (2007). *Ontology-driven information systems: Challenges and requirements*. In International Conference on Semantic Web and Digital Libraries. Indian Statistical Institute Platinum Jubilee Conference Series (2007) 35–44.

(Yu, Liu & Li, 2001)

Yu, E. Liu, L. & Li, Y. (2001). *Modelling Strategic Actor Relationships to Support Intellectual Property Management.* In H. S.Kunii, S. Jajodia, & A. Sølvberg (Eds.), Conceptual Modeling — ER 2001 (pp. 164–178). Springer Berlin Heidelberg. Available online at http://link.springer.com/chapter/10.1007/3-540-45581-7_14.

(Zave, 1997)

Zave, P. (1997). *Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys,* 29(4): 315-321.

(Zowghi & Coulin, 2005)

Zowghi, D., & Coulin, C. (2005). *Requirements Elicitation: A Survey of Techniques, Approaches, and Tools.* In A. Aurum & C. Wohlin (Eds.), Engineering and Managing Software Requirements (pp. 19–46). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/3-540-28244-0_2

AUTHORS PUBLICATIONS

Stynes, P., Conlan, O., O'Sullivan, D. (2009), Supporting Organisational Change through Enhancing Shared Understanding and Simulated Infrastructure Modelling. IN: Proceedings of the PhD Colloquium held at the Winter Simulation Conference, 2009, 13th-16th December, Austin, Texas, USA.

Stynes, P., Conlan, O., O'Sullivan, D. (2009) Simulation-based Communication Tool: an Enabler for Collaborative Decision Making. IN: Proceedings of the Workshop on Controlled Natural Language (CNL) 2009, 8th-10th June, Marettimo Island, Italy.

Stynes, P., Conlan, O., O'Sullivan, D. (2008) Towards a Simulation-based Communication Tool to Support Semantic Business Process Management. IN: Proceedings of the Fourth International Workshop on Semantic Business Process Management in Proceedings of Workshops held at the Fifth European Semantic Web Conference, (ESWC08), 2nd June, Tenerife, Canary Islands, Spain.

Stynes, P. & Conlan, O. (2006) Architectural Framework for the Composition and Delivery of Adaptive Educational Support Services. IN: Proceedings of the International Workshop on Applying Service Oriented Architectures to Adaptive Information Systems in Proceedings of Workshops held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), 20th June, Dublin. Dublin, National College of Ireland. pp. 156-163.

APPENDICES

Appendix 1.   Experimental Study 1 Trial Documents

Instructions for completing the questionnaire

Hi ,

Many thanks for agreeing to participate in the usability study of a system that allows the Head of School describe an organizational goal and rules in natural language and simulate the change to the organization.

Please find enclosed a PowerPoint presentation that creates a scenario of using the system and an evaluation survey.

I would be grateful if you could go through the PowerPoint presentation, complete the survey and then return the survey to me at pstynes@ncirl.ie.

Scenario Based Prototype

Simulation
of
Organizational Processes
for
Improved Decision Making

Paul Stynes

Purpose

- Allow a Head of School specify their organisational goal and rules in English
- The system will interpret the goal and identify the organizational process that can deliver the appropriate service
- The system will adapt the service based on the rules specified with the goal
- The Head of School can communicate with the system by modifying, adding or deleting new rules and then simulating the changes on the organisation
- The system will display the new changes to the organization and any organizational rules that are no longer valid.

Problem

- Optimize organizational changes using natural language from the users domain
- Allow the user make changes to the rules within an organisation and review the impact they have
- Simulate the interaction of the process with people in the organization

How

- Develop an interface that is grounded in the language of the Head of School
- Simulation of the resources in the organization and their interaction with the organizational process
- Create a rule base that is responsive to the Head of Schools changing needs.

Benefits

- Improved Decision Making by allowing the user make changes to organizational rules and then observe the impacts they have on the organization
- Optimize the use of resources in an organization
- Helps formulate changes that are required in an organization

Example

- The following example highlights
  - Scenario based on the scheduling organizational process
  - Sequence of screens representing screenshots of the Graphic User Interface

Scenario - Scheduling

- Goal
  - scheduling faculty to modules in classrooms.
- Current State
  - Faculty teach 6 hours
  - No equipment in any classroom
  - Room 3.01 contains seating for 50 students
- Mainflow
  - The **goal** of the head of school is to schedule the Bsc (Hons) in Software Systems.
  - One **rule** is to schedule all **faculty** to teach 14 hours.
  - Another rule is that all **classrooms** contain a multimedia projector.
  - Another example of a change in an organisation rule is a change in the size of a classroom. In this scenario room 3.01 will change to allow seating for 100 students.
- Results
  - Details of successful executions of the rules i.e. Faculty teaches 14 hours and the modules they will teach such as computer architecture etc.
  - The issues highlight the rules that did not fulfil the objectives. The Head of School has a choice of accepting these issues or modifying the rules. An example is where a faculty member is only scheduled for 6 hours
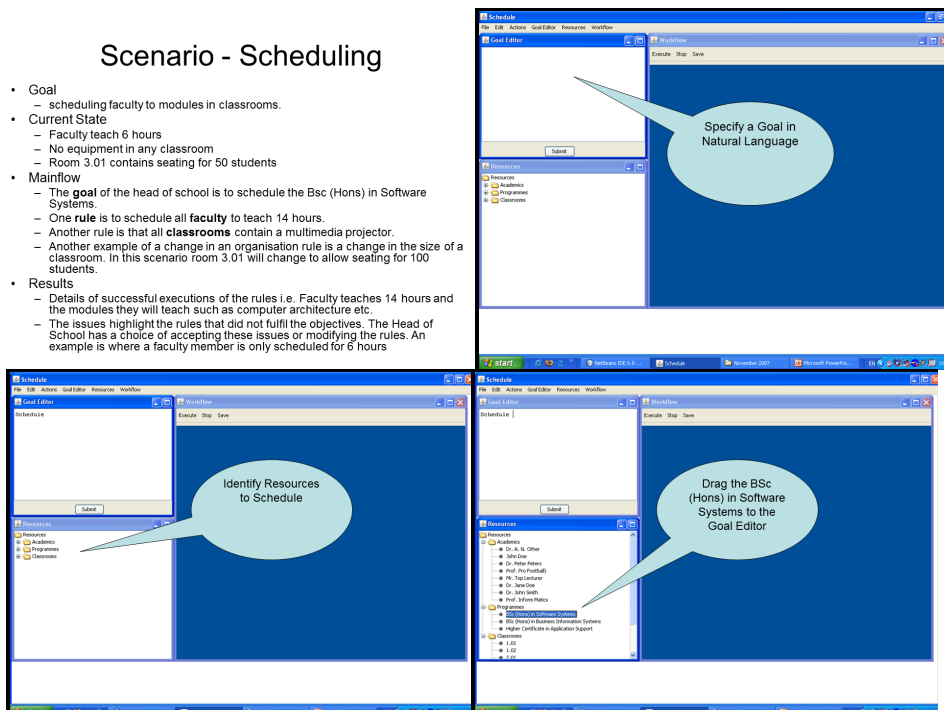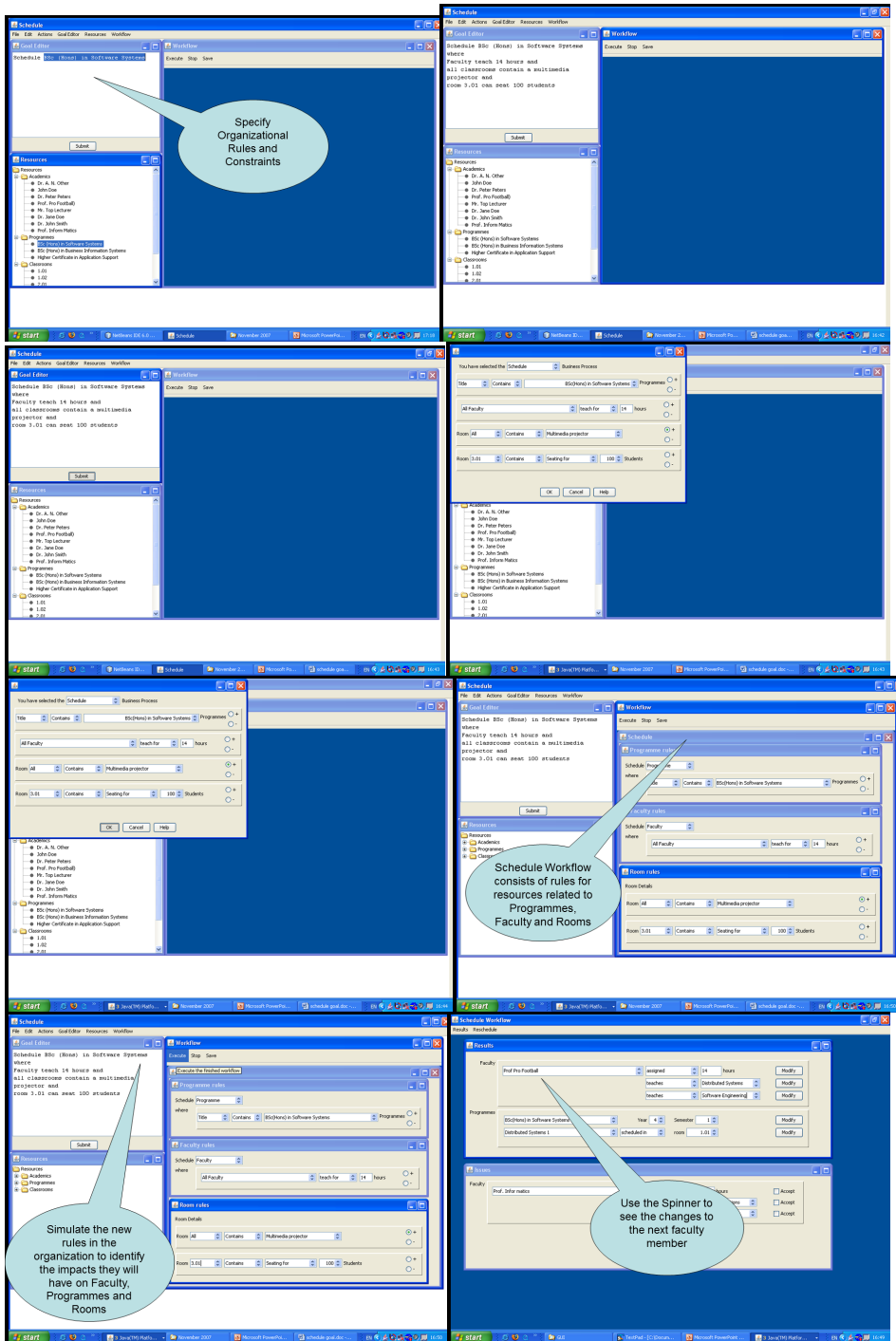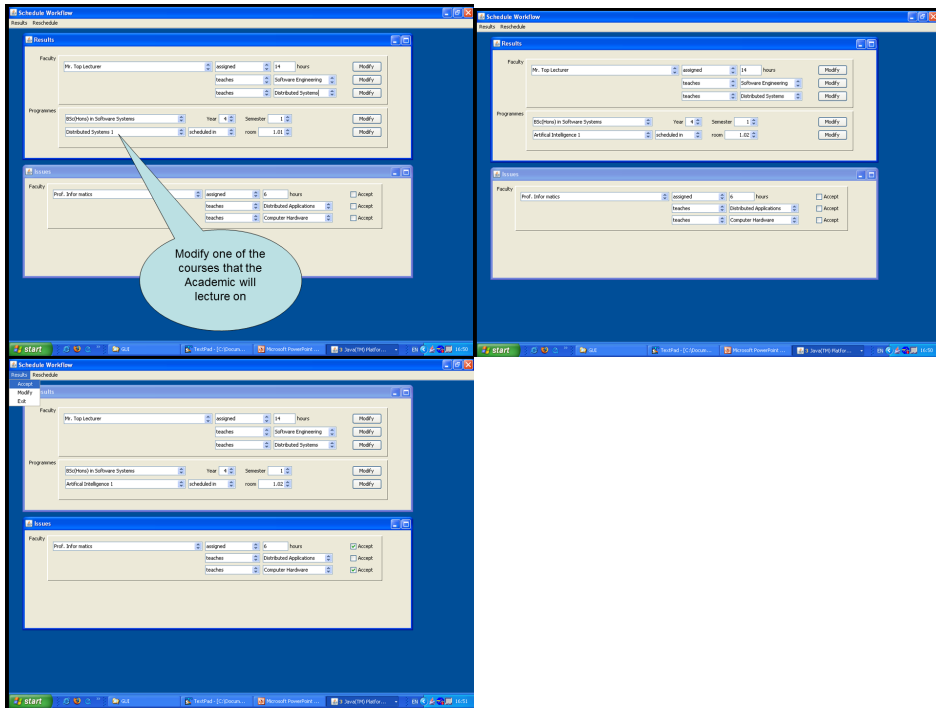
Interface Systems Usability Questionnaire

Demographics

1 What is your name?

2 What is your gender?   Male   Female

3 What is your role in work

Perceived Usefulness   1 2 3 4 5

4 Using the system in my job would increase the organisations productivity   Unlikely | | | | | | Likely

5 This system would lead to quicker responsiveness in the organization i.e. lead to an agile organization   Unlikely | | | | | | Likely

Usability Heuristics   1 2 3 4 5

6 Simple and Natural Dialogue   bad | | | | | | Good

User Interface Satisfaction

7 Overall Reaction to the Software   terrible | | | | | | Wonderful

8 Overall Reaction to the Software   difficult | | | | | | Easy

Screen   1 2 3 4 5

9 Sequence of screens   confusing | | | | | | very clear

Learning   1 2 3 4 5

10 Performing tasks is straightforward   never | | | | | | Always

Project specific questions   1 2 3 4 5

11 The Schedule scenario is understandable   I strongly agree | | | | | | I strongly disagree

12 The rules for Programmes, Faculty and Classrooms are understandable.   I strongly agree | | | | | | I strongly disagree

13 The rules for Programmes, Faculty and Classrooms are desirable   I strongly agree | | | | | | I strongly disagree

14 Do you think the organisational rules are useful to a scheduling application   I strongly agree | | | | | | I strongly disagree

15 The use of a Goal editor for entering an organizational goal in natural language is desirable   I strongly agree | | | | | | I strongly disagree

16 The resource tree is helpful in formulating an organizational goal   I strongly agree | | | | | | I strongly disagree

17 The systems automatic interpretation of the goal in terms of business process and rules is desirable.   I strongly agree | | | | | | I strongly disagree

18 The interation with the system to add/modify/delete new rules is desirable   I strongly agree | | | | | | I strongly disagree

19 The display showing results related to organizational goals that were successfully changed is desirable   I strongly agree | | | | | | I strongly disagree

| 20 | The display highlighting issues that did not fulfil the organizational goals is desirable | I strongly agree | | | | | | I strongly disagree |

Images

Spinner   **Combo box**   **List Box**

BSc(Hons) in Software Systems ⬍

| BSc (Hons) IN Software Systems | ⌄ |
| BSc (Hons) IN Software Systems |
| BSc (Hons) in Business Information Systems |
| Higer Certificate in Application Support |
| All |

| 21 | The scenario used spinners for displaying options and results. From the images above, which do you think is more appropriate to use for the display of options and results i.e. Spinners Combo box or a List Box | _____ |

List the most negative aspects

22

23

24

List the most positive aspects

25

26

27

List any suggested improvements

28

29

List any scenarios that are more relevant to your job

30

Appendix 2.   Experiment 2 Trial Documents

Domain Understanding Interview Questions

(i)   What is the process for identifying the software requirements for creating a software environment in a) computer room or b) lecturers laptop at the National College of Ireland?

(ii)   What is your opinion about the length of time that it takes to communicate the software requirements?

(iii)   Do requirements change after the process is complete and the software environment is set-up? If yes could these changes have been planned during the process? If yes how do you feel about these changes?

(iv)   What are the causes that lead to changes in the software environment?

(v)   Does the process lead to collaboration about creating a software environment between you and the people that request the software? Where collaboration could involve analysing the cost of software, rejecting the software, purchasing other software. If yes could you describe the collaboration?

(vi)   What is the positive aspect of this approach?

(vii)   What is the negative aspect of this approach?

(viii)   How can this process be improved?

(ix)   (Wrap up question) Who should I turn to, to learn more about this topic?

Invite to level of service and qualification online questionnaire

Dear Paul,

I would like to invite you to complete the first phase of the study. This phase involves answering questions about the length of service, experience and qualification. In some questions example answers are provided.

This part will take approximately 3 minutes and can be completed at http://crilt.ncirl.ie/limesurvey/index.php?sid=92363&lang=en.

Yours sincerely,

Paul Stynes

…

Pre-Test - Level of Service and Qualifications Survey

This survey identifies the demographics, level of service and qualifications of subjects in the study.

Many thanks for agreeing to participate in this survey.

This survey is part of the first phase of the study where you will answer questions about demographics, the length of service with NCI and other organisations and your qualifications. This part will take approximately 3 minutes.

 The objective of the overall study is to investigate the use of asynchronous tools to aid communication in developing a shared representation of IT support changes. These IT support changes should aid the necessary collaborative decision making in support of the organisation's goals.

There are 9 questions in this survey

Demographic Details

1   What is your Name? *

Please write your answer here:

2   What is your Gender? *

Please choose **only one** of the following:

- ⚪Female
- ⚪Male

3   What is your age? *

Please choose **only one** of the following:

- ⚪0 - 20
- ⚪21 - 25
- ⚪26 - 30
- ⚪31 - 35
- ⚪36 - 40
- ⚪41 - 45
- ⚪46 - 50
- ⚪>50

Service Details

The following questions identify details about your length of service and experience with your current organisation and your length of service with past organisations.

4   What is your title? *

Please write your answer here:

 Example

Lecturer

5   What is your role in the organisation? *

Please write your answer here:

 Example

The role involves lecturing in modules at undergraduate and post graduate level. The role involves managing student internet accounts.

6   Describe the work experience that you have gained in your current role *

Please write your answer here:

Example

I have gained experience at installing IP networks.

My experience involves lecturing at 3rd level.

7 **What is the number of years of service with NCI? ***

Please choose **only one** of the following:

- ○1 - 2 years
- ○3 - 4 years
- ○5 - 6 years
- ○7 - 8 years
- ○9 - 10 years
- ○>10 years

8 **What is the number of years of service prior to joining your present organisation? ***

Please choose **only one** of the following:

- ○1 - 2 years
- ○3 - 4 years
- ○5 - 6 years
- ○7 - 8 years
- ○9 - 10
- ○> 10 years

Qualification Details

The following questions identify the qualifications that you have obtained and their relevance to your experience.

9 **Describe your qualifications and certificates and indicate if they are relevant to your role. ***

Please write your answer here:

Example

B.Sc. (Hons) in Computer Science (Relevant).

Interconnecting CISCO Networking Devices Part 1 v1.0 (Relevant).

Certificate in Drama (Not relevant).

Invite to discuss organisational change scenario

Dear XXXX,

I would like to invite you to complete the second phase of the study on collaborative communication among team members during organisational change and the creation of an IT system to automate the change.

The second phase will involve a discussion on the organisational change scenario and the roles of each team member. This will take approximately 10 minutes and will take place at the following times in meeting room 3:

- Wednesday 4th March at 12:00
- Thursday 5th March at 3:00pm
- Friday 6th March at 10:00am

Could you please let me know the day and time that works best for you. I can make alternative arrangements if none of the above times are suitable.

Yours sincerely,

Paul Stynes

…

Organisational Change Scenario

The school of computing are in the process of changing programmes from the BSc (Hons) in Software Systems to the BSc (Hons) in Computing with specialisations in

1. Gaming and Multimedia Design
2. Networking and Mobile Technologies
3. Software Systems

The HETAC approved course schedule is described in Appendix 1 BSc (Hons) in Computing Programme Schedule.

The IT Department require assistance in identifying the new software and hardware requirements that they will roll out to all the computer labs in the National College of

Ireland. This platform will contain all the software and hardware requirements that support the BSc (Hons) in Computing. The computer labs contain computers and so there is no need to specify a computer as a hardware requirement.

Objective

The objective of the scenario is to

1. Communicate among your team members through SBCT and email to define the software and hardware requirements for each module
2. Create and populate an IT system that will allow academics store/update/delete the software and hardware requirements.

As part of the communication you are required to CC all emails regarding this scenario to sbctgroup3@gmail.com.

Executive Role

The executive's role is to identify and approve software that relates to the colleges strategy. Preference is giving to software for Microsoft Web Development technologies, Cisco Internet hardware and SAP database software.

…

IT Architect Role

The IT Architects role is to accept software and hardware that is reasonable priced. The IT Architect will collaborate with the group to identify software and hardware that can meet the requirements of each module. They can also reject software and hardware if it can be handled by other software/hardware applications. They can reject software and hardware if it is greater than 50,000 euros. The IT Architect is primarily responsible for identifying a suitable IT system for storing the data in collaboration with other members of the group.

…..

Staff Role

The academic (Staff) will represent all lecturers that are teaching the modules. Their role is to identify software and hardware that they would like to use in each module in collaboration with other team members. They can accept recommendations from other

members of the group or may reject software based on the criteria that it does not meet the teaching needs of the current curriculum or is not in line with their teaching practises for that module.

Appendix 1. BSc(Hons) in Computing Programme Schedule

Year 4					
Areas of Specialization			Gaming and Multimedia Design Specialization	Networking and mobile Technologies specialization	Software Systems Specializati
Sem 2	Software Project[1]	Strategy Management	Applied Artificial Intelligence	Distributed Systems	Distributed Systems
			Multimedia and Mobile Application Development	Multimedia and Mobile Application Development	Applied Artificial Intelligence
Sem 1	Software Project[1]	Change Management	Network Programming and Distributed Systems	Computer Graphics Design and Animation	Business and Network Security	Business a: Network Security

Year 3					
Sem 2	Work Placement				
Sem 1	Advanced Internet Technologies	Team Project[2]	Introduction to Artificial Intelligence	Wireless Networking	Advanced Programming

Year 2					
Sem 2	Object Oriented Software Engineering	Advanced Databases	Project[2]	Data Communications and Networking	Software Proje Management
Sem 1	Introduction to Software Engineering	Introduction to Databases	Web Development[2]	Data Structures and Algorithms	Business Entrepreneurship

Year 1					
Sem 2	Object Oriented Programming	Computer Architecture	Introduction to Multimedia[2]	Operating Systems	Managerial Foundations Information Systems
Sem 1	Introduction to Programming	Introduction to Computers	XHTML & Web Design[2]	Introduction to Mathematics for	Personal a: Professional

[1] Denotes a 15 credit module

209

Invitation for the workshop using the brainstorming technique

Dear XXX,

I would like to invite you to complete the third phase of the study. The third phase will involve a 30 minute session for the teams to brainstorm ideas on the concepts (types of data) that represents the organisational goals and sample data, the rules that govern the concepts and who approves the data that is entered. This phase involves participation by all three members of a team.

The third phase will take place at the following times in meeting room 3:

- Wednesday 11th March at 12:00
- Thursday 12th March at 3:00pm
- Friday 13th March at 11:15am

Could you please let me know what time suits? I can make alternative arrangements if none of the above times are suitable.

Yours sincerely,

Paul Stynes

Invitation to controlled experiment

Dear Paul,

I would like to invite you to complete the fourth phase of the study. The fourth phase will involve collaboration among the team members to identify and agree on the actual data based on the concepts defined in the third phase. This phase will also involve identifying and creating a suitable IT system for storing the data. This phase will take approximately 1 hour.

During this phase team members are only allowed to communicate through the Simulation Based Communication Tool (SBCT) and email to the other team members. You are not allowed to discuss any issues that relate to the research experiment with other members of the team. All email communication must include the following email address sbctgroup1@gmail.com.

The fourth phase will take place at the following times in the ICELT research room:

- Friday 27th February at 11:00
- Monday 2nd March at 9:00
- Monday 2nd March at 11:00
- Monday 2nd March at 2:00pm
- Monday 2nd March at 3:30pm
- Tuesday 3rd March at 9:00
- Wednesday 4th March at 12:00
- Thursday 5th March at 3:00pm

Could you please let me know what times suit? I can make alternative arrangements if none of the above times are suitable.

Yours sincerely,

Paul Stynes

Post-Test Questionnaire - Asynchronous Communication for Decision Making

This survey identifies the subjects' perception of how asynchronous tools aid communication in developing a shared representation of IT support changes that are needed and can aid the necessary collaborative decision making in support of an organisations goal.

There are 13 questions in this survey

Demographics

1   What is your name? *

Please write your answer here:

Frequency of Communication

The following questions identify the participant's perception of the groups and their frequency of communication.

2   Please estimate the number of messages that you sent. *

Please write your answer here:

3   Please estimate the number of messages that the group (including you) sent. *

Please write your answer here:

Quality of communication

The following questions identify the quality of messages that the group and you sent based   on
1) messages that suggest solutions

2) Messages that identify constraints on organisational changes

3) Messages that identify organisational issues but no solutions.

4 **Please estimate the number of messages that you sent that suggest a solution ***

Please write your answer here:

5 **Please estimate the number of messages that the group sent (including you) that suggest a solution ***

Please write your answer here:

6 **Please estimate the number of messages that you sent that identify constraints on organisational changes. ***

Please write your answer here:

7 **Please estimate the number of messages that the group (including you) sent that identify constraints on organisational changes. ***

Please write your answer here:

8 **Please estimate the number of messages that you sent that identify organisational change issues. ***

Please write your answer here:

9 **Please estimate the number of messages that the group (including you) sent that identify organisational change issues. ***

Please write your answer here:

Perception

The following statements identify your perception about
1) Collaborative communication
2) Shared representation of IT systems supporting organisational change
3) Collaborative decision making

10 Do you agree with the following statements? *

Please choose the appropriate response for each item:

	strongly disagree			strongly agree
The approach of communicating collaboratively increases communication.	○	○○○	○	
The group have a shared representation of the organisational change.	○	○○○	○	
The group have a shared representation of the IT system.	○	○○○	○	
The IT System accurately represents and specifies the organisational goals.	○	○○○	○	
The IT system supports organisational change faithfully	○	○○○	○	
Decisions were made in a collaborative way	○	○○○	○	
My understanding of IT infrastructure changes to support the organisations goals has increased.	○	○○○	○	
The groups understanding of IT infrastructure changes to support the organisations goals have increased.	○	○○○	○	

	strongly disagree				strongly agree
My understanding of IT infrastructure changes to support the organisations goals has increased due to collaboration.	○		○	○	○ ○
The groups understanding of IT infrastructure changes to support the organisations goals have increased due to collaboration.	○		○	○	○ ○
The proposed IT system has a fidelity to commercial systems and may be interchangeable.	○		○	○	○ ○

11 List any negative aspects of the experiment. *

Please write your answer here:

12 List any positive aspects of the experiment. *

Please write your answer here:

13 List any suggested improvements to the experiment. *

Please write your answer here:

Post Test Questionnaire - Simulation Based Communication Tool Survey

This survey identifies the usability characteristics of the Simulation Based Communication Tool.

Many thanks for agreeing to participate in this survey.

The objective of the overall study is to investigate the use of asynchronous tools to aid communication in developing a shared representation of IT support changes that can aid the necessary collaborative decision making in support of the organisation's goals.

This survey is part of the fourth phase of the study where you will answer questions about demographics, perceived usefulness of the Simulation Based Communication Tool, usability heuristics, user interface satisfaction, screen layout, learning and project specific questions.

This part will take approximately 5 minutes.

There are 11 questions in this survey

Demographic Details

1 What is your name? *

Please write your answer here:

Perceived Usefulness

2 Do you agree with the following statements? *

Please choose the appropriate response for each item:

	strongly disagree			strongly agree
Communication increases using the Simulation Based	○	○	○	○ ○

	strongly disagree					strongly agree

Communication Tool compared to other forms of asynchronous communication like email

Communication that involves identifying solutions to organisational changes increases using the Simulation Based Communication Tool

	strongly disagree					strongly agree
	○		○	○	○	○

Using the Simulation Based Communication Tool helps to identify constraints on organisational changes

	○		○	○	○	○

Using the Simulation Based Communication Tool helps the group create a shared representation of how IT can support the organisational changes

	○		○	○	○	○

Using the Simulation Based Communication Tool helps the group make collaborative decisions

	○		○	○	○	○

The Simulation Based Communication Tool's simulation of the web service architecture supports the organisational change faithfully (where faithful (marked by fidelity to an original) "a close translation"; "a faithful copy of the portrait"; "a faithful rendering of the observed facts")

	○		○	○	○	○

Usability Heuristics

3   Simple and Natural Dialogue *

Please choose the appropriate response for each item:

	bad				good
Simple and Natural Dialogue	○	○	○	○	○

217

User Interface Satisfaction

4   Please rate the following statement *

Please choose the appropriate response for each item:

	terrible				wonderful
Overall reaction to the software	○	○	○	○	○

5   Please rate the following statement *

Please choose the appropriate response for each item:

	difficult				easy
Overall reaction to the software	○	○	○	○	○

Screen Layout

6   Please rate the following statements *

Please choose the appropriate response for each item:

	very confusing				very clear
Sequence of screens	○	○	○	○	○
The screen that appears when selecting the Rules button	○	○	○	○	○
The Filter Rules drop down list and the update to the Rules Base	○	○	○	○	○
The Controlled Natural Language Editor screen that appears when selecting the Create Rule Button	○	○	○	○	○
Entering rules based on subject predicate object e.g.	○	○	○	○	○

	very confusing		very clear

IntroductionToProgramming requires Netbeans.

The screen that appears when selecting the Collaborate button	○	○○○	○
Collaboration by selecting a service and completing the service details	○	○○○	○
Collaboration by selecting a service and completing the service details	○	○○○	○
The screen that appears when selecting the Approve Changes button	○	○○○	○
The screen that appears when selecting the Approve Changes button	○	○○○	○
Selecting a rule and approving that rule	○	○○○	○
Knowing who originated the rule that requires approval	○	○○○	○
Creating the IT Architecture	○	○○○	○
The screen that appears when selecting a web service tester	○	○○○	○
The screen that appears when selecting a web service interface (wsdl file)	○	○○○	○
The screen that appears when selecting the data associated with a web service	○	○○○	○

Learning

7 Please rate the following statement *

Please choose the appropriate response for each item:

	never			always	
Performing tasks is straightforward	○	○	○	○	○

Project Specific Questions

8 Do you agree with the following statements *

Please choose the appropriate response for each item:

	strongly disagree			strongly agree	
The organisational change scenario is understandable	○	○	○	○	○
The rules for the organisational change scenario are understandable	○	○	○	○	○
The use of a Controlled Natural Language Editor for creating the rules is desirable	○	○	○	○	○
The Simulation Based Communication Tools automatic interpretation of the organisational rules to create the Web Service architecture is desirable	○	○	○	○	○
The interaction with the Simulation Based Communication Tool to create rules is desirable	○	○	○	○	○
The use of a filter for viewing the rule base is desirable	○	○	○	○	○
The use of a collaboration facility for decision making is desirable	○	○	○	○	○

9   What did you like about the Simulation Based Communication Tool? *

Please write your answer here:

10 What did you dislike about the Simulation Based Communication Tool? *

Please write your answer here:

11 What can be improved with the Simulation Based Communication Tool? *

Please write your answer here:

Appendix 3.   Experiment 3 Trial Documents

TRINITY COLLEGE DUBLIN

INFORMATION SHEET FOR PARTICIPANTS

Background of Research:

CRESUS (Collaborative Requirements Elicitation through enhancing Shared Understanding and Simulation) addresses the need for a technical architecture framework that supports the collaboration among stakeholders during the communication of the organisation's requirements in a semantically consistent and understandable manner and then reflecting the potential impact of those requirements on the IT infrastructure of a business process.

The purpose of this study is to evaluate the effectiveness of the Communication and Simulation Tool, CRESUS-T at supporting collaborative requirements elicitation through enhancing the shared understanding of organisational and associated IT infrastructure requirements around communication and business process simulation controlling for level of service, academic qualifications, and characteristics of subjects.

Procedures of this Research:

- You will be asked to fill in questionnaires before, during and after the experiment [1/2 hour]
- An administrator will deliver a training session on the Business Process Modelling Notation (BPMN) followed by a questionnaire that examines your level of understanding of BPMN. [1 hour].
- You will receive a description of the problem domain that represents some information provided during the early stages of requirements elicitation. Your task is to communicate asynchronously with other stakeholders in your team to develop a shared understanding of the IT systems that will support the business process of the problem domain. This includes identifying any data stored in the system. You will be able to develop this understanding through the collaborative creation of a business process diagram, prototype IT system and data models. This part of the experiment will run over a one week period in a natural setting and in conjunction with your other work responsibilities. You will be requested to work on the experiment every day for at least 10 minutes between the hours of 10:00 and 12:00. [At least 50 minutes]

- You will be asked to keep a journal during the experiment.
- You will be observed during the experiment.
- Data will be logged during the experiment.
- You will be interviewed on questions that relate to the experiment and to clarify and data collected during the experiment. The interview will be recorded. [1/2 hour]
- You will be debriefed on the experiment. [30 minutes]
- The total duration of the study should be approximately 3.5 hours over a 1 week period.

Participant Selection Procedure:

- Participants will be selected from the School of Computing and IT Department.

Health and Safety:

- If you have a history of Photosensitive Epilepsy or any other condition that may cause you to experience difficulty with this experiment, please inform the supervisor of the study.

Conflict of Interest:

- Please be advised that this research is being conducted by an employee of the college which may involve taking advantage of an existing relationship in order to progress this research.

Legal Notes:

- All information collected for this study will be recorded anonymously and held confidentially to protect your privacy. All information gathered will be used solely for this study.
- Any data used in publications will be aggregated and no personally identifiable information we revealed.
- Participation in this study is voluntary; you may withdraw from the study at any time for any reason and omit questions you do not wish to answer without penalty.
- In the extremely unlikely event that illicit activity is reported to me during the study I will be obliged to report it to appropriate authorities.

If you would like further details about the study feel free to contact Paul Stynes using the details below.
Name: Paul Stynes   **Email:** pstynes@ncirl.ie   **Tel:** 01 449 8613

Address: National College of Ireland, Mayor Street, Dublin 1.

TRINITY COLLEGE DUBLIN

INFORMED CONSENT FORM

LEAD RESEARCHERS:

Paul Stynes

BACKGROUND OF RESEARCH:

CRESUS (Collaborative Requirements Elicitation through enhancing Shared Understanding and Simulation) addresses the need for a technical architecture framework that supports the collaboration among stakeholders during the communication of the organisation's requirements in a semantically consistent and understandable manner and then reflecting the potential impact of those requirements on the IT infrastructure of a business process.

The purpose of this study is to evaluate the effectiveness of the Communication and Simulation Tool, CRESUS-T at supporting collaborative requirements elicitation through enhancing the shared understanding of organisational and associated IT infrastructure requirements around communication and business process simulation controlling for level of service, academic qualifications, and characteristics of employees at the National College of Ireland.

PROCEDURES OF THIS STUDY:

- You will be asked to fill in questionnaires before, during and after the experiment [1/2 hour]
- An administrator will deliver a training session on the Business Process Modelling Notation (BPMN) followed by a questionnaire that examines your level of understanding of BPMN. [1 hour].
- You will receive a description of the problem domain that represents some information provided during the early stages of requirements elicitation. Your task is to communicate asynchronously with other stakeholders in your team to develop a shared understanding of the IT systems that will support the business process of the problem domain. This includes identifying any data stored in the system. You will be able to develop this understanding through the collaborative creation of a business process diagram, prototype IT system and data models. This part of the experiment will run over a one week period in a natural setting and in conjunction with your other work responsibilities. You will be requested

to work on the experiment every day for at least 10 minutes between the hours of 10:00 and 12:00. [At least 50 minutes]
- You will be asked to keep a journal during the experiment.
- You will be observed during the experiment.
- Data will be logged during the experiment.
- You will be interviewed on questions that relate to the experiment and to clarify and data collected during the experiment. The interview will be recorded. [1/2 hour]
- You will be debriefed on the experiment. [30 minutes]
- The total duration of the study should be approximately 3.5 hours over a 1 week period.

PUBLICATION:

It is intended that results of this study will be published in Paul Stynes's Ph.D. thesis and a relevant scientific journal. Individual results will be aggregated anonymously and research reported on aggregate results.

DECLARATION:

- I am 18 years or older and am competent to provide consent.
- I have read, or had read to me, a document providing information about this research and this consent form. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction and understand the description of the research that is being provided to me.
- I agree that my data is used for scientific purposes and I have no objection that my data is published in scientific publications in a way that does not reveal my identity.
- I understand that if I make illicit activities known, these will be reported to appropriate authorities.
- I understand that I may stop electronic recordings at any time, and that I may at any time, even subsequent to my participation have such recordings destroyed (except in situations such as above).
- I understand that, subject to the constraints above, no recordings will be replayed in any public forum or made available to any audience other than the current researchers/research team.
- I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights.
- I understand that I may refuse to answer any question and that I may withdraw at any time without penalty.
- I understand that my participation is fully anonymous and that no personal details about me will be recorded.
- I understand that if I or anyone in my family has a history of epilepsy then I am proceeding at my own risk.
- I understand that this research is being conducted by an employee of the college which may involve taking advantage of an existing relationship in order to progress this research.
- I have received a copy of this agreement.

PARTICIPANT'S NAME: _____

PARTICIPANT'S SIGNATURE: _____ **Date:**
_____

Statement of investigator's responsibility: I have explained the nature and purpose of this research study, the procedures to be undertaken and any risks that may be involved. I have offered to answer any questions and fully answered such questions. I believe that the participant understands my explanation and has freely given informed consent.

RESEARCHERS CONTACT DETAILS:

Name: Paul Stynes   **Email:**pstynes@ncirl.ie   **Tel:** 01 449 8613

Address: National College of Ireland, Mayor Street, Dublin 1.

INVESTIGATOR'S SIGNATURE: _____ **Date:**
_____

Case Study – Roll out of the IT Software Image

The IT Department of the National College of Ireland (NCI) undergo a business process for rebuilding a software image on an annual basis. The software image is then put on all desktop computers in the computer laboratories in NCI. The software image contains the software that is required for each module on all programmes offered by NCI.

An IT administrator is responsible for initiating the business process when they send an email to all lecturers in each school around March or April. The email contains the previous year's software requirement list. The software requirement list contains details about the software application and its version number. Lecturers in each school have four to six weeks where they can review the software that they require for the next academic year.

The Lecturers will check each module's current IT requirements. They will then review the module software requirements for the next academic year. The lecturer's module software requirements are compared to the software requirement list. If there are new software requirements or changes to existing versions of software then the lecturer can update the module software requirements and the software requirement list with the software that is required for their module for the next academic year. For open source software, the lecturer has to also download the software to a global directory. The updated software requirement list is returned to the IT administrator.

However the software requirement list never comes back on the due date. IT sends a reminder email at least three times and sometimes it is escalated to the heads of school.

The IT administrator starts to build the software image around the end of May or beginning of June. To build a Citrix server farm from the ground up could take six weeks. This involves taking one server out of a live environment. The server is wiped of software, performing a raid and mirroring, followed by an operating system, Microsoft patches, Citrix patches, and then the software image based on the software identified from the software requirements list. In addition, different policies are applied and scripts are executed. There may be some compatibility issues for example PhPDev cannot run on Citrix. The issue is around running a server on a server. Tests are carried out on the build to ensure that all the software works and that the software is compatible. If the software doesn't work then it has to be tested in isolation. The computer laboratories are

upgraded over the years and as a result there will be different hardware platforms that have to be tested with the software. For example one lab may contain Dell GX520 and 270's in another lab. The image is then rolled out to all the Desktop computers in the computer laboratory.

The email that IT sends around March or April emphasises that this is the one and only one time that a request for software can be made. Installing software outside of the software image doesn't work well. The problem that may arise is when a virus is detected on a pc in the computer laboratory. That computer laboratory could be quickly re-imaged. However software outside of the software image would be lost. Occasionally exceptions do occur where a lecturer may require a piece of software for 10 or 15 students in their class. The software can be installed on the first two rows of a certain lab under the provision that should something happen in the lab and the computers are re-imaged, then the software will be lost. The installation of additional software is a man-intensive activity, where several people form the IT Department will go down and install the software on each machine. The students that will then use that application have to be set up with the necessary privileges to run the software.

Programmes that require an IT environment

The course directors in the School of Computing are creating new conversion programmes in the areas of

4. Software Development
5. Cloud Computing
6. Web Development
7. Cloud Infrastructure
8. Data Analytics
9. Mobile and Cloud Gaming

The programme structure is described in **Error! Reference source not found.**.

The IT Department require assistance in identifying the new software and hardware requirements that they will roll out to all the computer labs in the National College of Ireland. This platform will contain all the software and hardware requirements that support the Higher Diploma.

Objective

The purpose of the scenario is to gain an understanding of the problem domain and gather the requirements of an IT infrastructure that supports the business process as described in the case study. This purpose can be achieved through the following 3 objectives

The first objective is to create a data model of the system. The data model should describe the structure of the data and the actual data need to support the system. Samples of data modelling techniques that may be used are:

> a. Entity-Relationship Diagrams
> b. Ontology
> c. Data Dictionary

The second objective is to create and populate an IT infrastructure that will support the business process described in the case study.

The third objective is to create a basic business process model that integrates the IT infrastructure created in the previous step. The model should demonstrate a variety of features of the BPMN notation such as pools, process, events, tasks, and gateways for decisions. There are several business process diagram techniques that may be used, such as:

> d. Business Process Model and Notation
> e. Activity Diagram
> f. Flowchart

Role

Participants will be assigned to various stakeholder roles during the Requirements Elicitation activity namely that of the Business Executive, IT architect and User of the system.

Executive

The executive's role is to identify and approve software that relates to the colleges strategy. Preference is giving to software for Microsoft Web Development technologies, Cisco Internet hardware and SAP software.

IT Architect

The IT Architects role is to accept software and hardware that is reasonable priced. The IT Architect will collaborate with the group to identify software and hardware that can meet the requirements of each module. They can also reject software and hardware if it can be handled by other software/hardware applications. They can reject software and hardware if it is greater than 50,000 euros. The IT Architect is primarily responsible for identifying a suitable IT system for storing the data in collaboration with other members of the group.

User

The academic will represent all faculty. Their role is to identify software and hardware that they would like to use in each module in collaboration with other team members. They can accept recommendations from other members of the group or may reject software based on the criteria that it does not meet the teaching needs of the current curriculum or is not in line with their teaching practises for that module.

Collaborative communication

You are required to communicate with your group asynchronously in order to create the data model, IT infrastructure and business process model of the case study above. Communication should take place through the CRESUS-T tool located at

http://54.238.49.81:8081/CRESUS/

The following table highlights the details such as role, login name for the Journal and contact email address of each person in your group for this study.

Stakeholder	Role	Login name	Password	email
Cristina Muntean	Business Executive	cmuntean	Participant id	cresus.be2@gmail.com
Eugene O'Loughlin	User	eoloughlin	Participant id	cresus.user5@gmail.com
Jonathan McCarthy	User	jmccarthy	Participant id	cresus.user3@gmail.com
Eamon Nolan	User	enolan	Participant id	cresus.user7@gmail.com
Neha Theti	User	ntheti	Participant id	cresus.user14@gmail.com
Michael Bradford	User	mbradford	Participant id	cresus.user11@gmail.com
Frances Sheridan	User	fsheridan	Participant id	cresus.user12@gmail.com
Derek Caprani	User	dcaprani	Participant id	cresus.user13@gmail.com
Robert Duncan	IT Architect	rduncan	Participant id	cresus.ita1@gmail.com

Journal

You are required to keep a daily log that describes two highlights and two challenges that occurred during the study. Please use the login name and password from the previous table to login. The Journal is available at

http://54.238.49.81:8084/Journal/

Training

There is a serious of training courses available to help with the use of the CRESUS-T tool during data entry, IT infrastructure and BPMN modelling. These tutorials may be viewed at

- CRESUS Approach part 1 http://youtu.be/uMnfl-bF10k
- CRESUS-T Data Entry Part 2 http://youtu.be/uiJMEZNXTfY
- CRESUS Approach IT Infrastructure Part 3 http://youtu.be/iPDDvTIRLyY
- CRESUS Approach BPMN Model Part 4 http://youtu.be/tpvgQCZJCh

Appendix 1 Programme Structure

Area of Specialisation			Software Development Specialization	Cloud Computing Specialisation	Web Development Specialisation	Infrastructure Specialisation	Data analytics Specialisation	Mobile and Cloud Gaming Specialisation
Semester 3			Industrial Placement Programme					
Semester 2	Project	Career Bridge	Web Services and API Development	Cloud Application Development	Designing for the web (UI, UX, IA, IxD)	Virtualisation	Social Media Analytics	Cloud Gaming
			Advanced Programming	Cloud Computing	Web Programming	Infrastructure Management	Business Intelligence	Graphics and Animation for Devices
			Datastructures and Algorithms	Web Services and API Development	Web Architectures	Data Storage and Management	Business Data Analysis	Mobile Game Development
Semester 1		Object Oriented Software Engineering	Software Development		Web Design	Computer Architecture, Operating Systems and Networks	Databases	

232

CRESUS Pre-test Online Questionnaire

The main objective of this survey is to obtain information about participants: -
- length of service in industry
- level of academic qualifications
- experience with the problem domain
- experience with modelling languages

this information is required to determine the allocation of participants to groups during the experiment.

TRINITY COLLEGE DUBLIN
INFORMATION SHEET FOR PARTICIPANTS

Welcome Message for this questionnaire:
Many thanks for agreeing to participate in this survey.
The main objective of this survey is to attain information about your
- length of service in industry
- level of academic qualifications
- experience with the problem domain
- experience with modelling languages
this survey will take approximately 3 minutes.

Special Instructions:
- Participation in this study is voluntary; you may withdraw from the study at any time for any reason and omit questions you do not wish to answer without penalty.
- Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.
- Please do not name third parties in any open text field of the questionnaire. Any such replies will be anonymised.

Background of Research:

CRESUS (Collaborative Requirements Elicitation through enhancing Shared Understanding and Scenarios) addresses the need for a requirements elicitation approach that supports the collaboration among stakeholders during the communication of the organisation's requirements in a semantically consistent and understandable manner and then reflecting the potential impact of those requirements on the IT infrastructure of a business process.

The purpose of this study is to evaluate the effectiveness of the Communication and Modelling Tool, CRESUS-T at supporting collaborative requirements elicitation through enhancing the shared understanding of organisational and associated IT infrastructure requirements around communication and business process modelling controlling for level of service, academic qualifications, and characteristics of subjects such as domain and modelling knowledge.

Procedures of this Research:

- You will be asked to fill in questionnaires before and after the experiment [10 minutes]
- An administrator will deliver a training session on the Business Process Modelling Notation (BPMN) followed by a questionnaire that examines your level of understanding of BPMN. [20 minutes].
- You will receive a description of the problem domain that represents some information provided during the early stages of requirements elicitation. Your task is to communicate asynchronously with other stakeholders in your team to develop a shared understanding of the IT infrastructure that will support the business process of the problem domain. This includes identifying any data stored in the system. You will be able to develop this understanding through the collaborative creation of a business process model, Stub IT implementation (prototype IT system) and data models. This part of the experiment will run over a one week period in a natural setting and in conjunction with your other work responsibilities. You will be requested to work on the experiment every day for at least 10 minutes. [At least 50 minutes]
- You will be asked to keep a journal during the experiment.
- You will be observed during the experiment.
- Data will be logged during the experiment.
- You will be interviewed on questions that relate to the experiment and to clarify any data collected during the experiment. The interview will be recorded. [15 minutes]
- You will be debriefed on the experiment. [15 minutes]

- The total duration of the study should be approximately 1.75 hours over a 2 week period.

Procedure:
• Participants will be selected from the School of Computing and IT Department.
Health and Safety:
• If you have a history of Photosensitive Epilepsy or any other condition that may cause you to experience difficulty with this experiment, please inform the supervisor of the study.
Conflict of Interest:
• Please be advised that this research is being conducted by an employee of the college which may involve taking advantage of an existing relationship in order to progress this research.

Legal Notes:
• All information collected for this study will be recorded anonymously and held confidentially to protect your privacy. All information gathered will be used solely for this study.
• Any data used in publications will be aggregated and no personally identifiable information we revealed.
• Participation in this study is voluntary; you may withdraw from the study at any time for

any reason and omit questions you do not wish to answer without penalty.
• Please do not name third parties in any open text field of the questionnaire. Any such replies will be anonymised.
• In the extremely unlikely event that illicit activity is reported to me during the study I will be obliged to report it to appropriate authorities.

If you would like further details about the study feel free to contact Paul Stynes using the details below.
Name: Paul Stynes Email:pstynes@ncirl.ie Tel: 01 449 8613 Address: National College of Ireland, Mayor Street, Dublin 1.

There are 9 questions in this survey

Participants Identification

1   Please enter the code assigned to you.

Please write your answer here:

Level of Service

The following questions identify details about your length of service with the National College of Ireland and your length of service with past organisations.

2   What is your job title in National College of Ireland (NCI)?

Please write your answer here:

Examples
Lecturer II
Associate Faculty
Senior IT Administrator

3   What roles do you fulfil at NCI?

Please write your answer here:

Examples
Course Director
Lecturer
IT Administrator

4   Number of years of service in NCI?

Please choose **only one** of the following:

- ○1-2 years
- ○3-4 years
- ○5-6 years
- ○7-8 years
- ○9-10 years
- ○>10 years

5 Number of years that you were in employment before joining NCI?

Please choose **only one** of the following:

- ○1-2 years
- ○3-4 years
- ○5-6 years
- ○7-8 years
- ○9-10 years
- ○> 10 years

Qualification Details

The following questions identify the qualifications that you have obtained and their relevance to your experience.

6 Highest level of award obtained on the National Framework of Qualifications (NFQ)?

Please choose **only one** of the following:

- ○Level 1 Certificate
- ○Level 2 Certificate
- ○Level 3 Certificate / Junior Certificate
- ○Level 4 Certificate / Leaving Certificate
- ○Level 5 Certificate / Leaving Certificate
- ○Level 6 Advanced Certificate / Higher Certificate
- ○Level 7 Ordinary Bachelor Degree
- ○Level 8 Honours Bachelor Degree / Higher Diploma
- ○Level 9 Master's Degree / Post-Graduate Diploma
- ○Level 10 Doctorate Degree / Higher Doctorate

Domain and Modelling Experience

The following questions identify the - knowledge of the domain (i.e. process for deploying the IT image) - knowledge of the modelling notation

7 Rate your knowledge on Deployment of the IT image as for example in NCI.

Please choose **only one** of the following:

- ○I have no knowledge
- ○I have poor knowledge
- ○I have average knowledge
- ○I have good knowledge
- ○I have expert knowledge

8 Rate your confidence at using the Business Process Model and Notation (BPMN).

Please choose **only one** of the following:

- ○not confident
- ○somewhat confident
- ○average confidence
- ○confident
- ○very confident

9 Rate your confidence at using other modelling languages such as Data Flow Diagrams, UML etc.

Please choose **only one** of the following:

- ○not confident
- ○somewhat confident
- ○average confidence
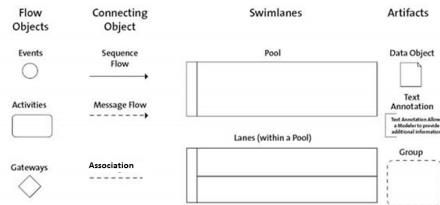- ○confident
- ○very confident

Thank you for completing this survey. Your contribution is valuable to this research.
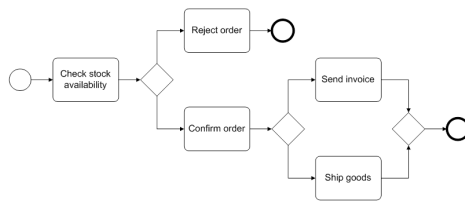
BPMN Training

Business Process Model and Notation

Notation

- A BPMN process model is a graph consisting of four types of elements (among others):

Order Management Process in BPMN First Try

Exclusive Gateways (XOR)

- Exclusive Decision / Merge
  - Indicates locations within a business process where the sequence flow can take two or more alternative paths.
  - **Only one** of the paths can be taken.
  - Depicted by a diamond shape that *may* contain a marker that is shaped like an "X".
  - Exclusive merge
    Proceed when one branch has completed
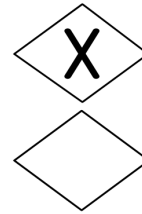
Parallel Gateway (AND)

- Parallel Fork / Join
  - Provide a mechanism to synchronize parallel flow and to create parallel flow.
  - Depicted by a diamond shape that *must* contain a marker that is shaped like a plus sign.
  - Parallel Fork
    takes all branches
  - Parallel join
    proceed when all incoming branches have completed
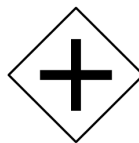
Revised Order Management Process
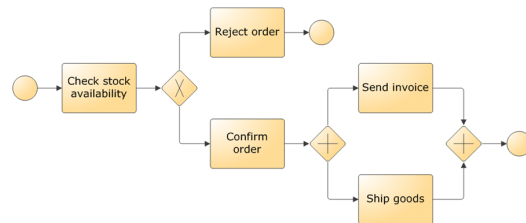
Collaboration - Order Management Process with Pools
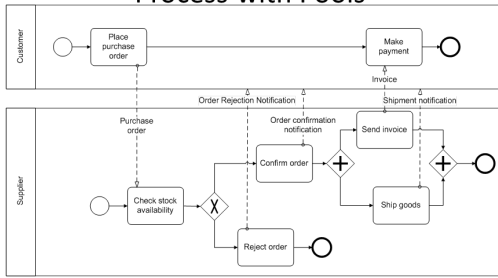
Order Management Process with Lanes
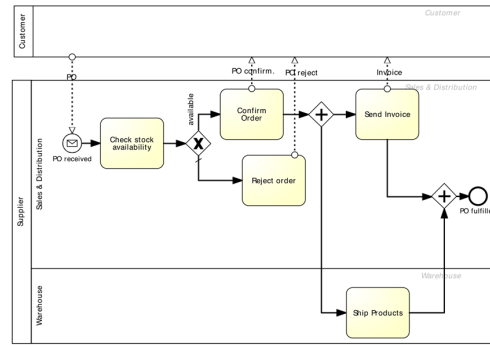
Order Processing Model with Artifacts

Additional Resources

- Introduction to BPMN v2.0 by Dr. Jim Arlow
  - http://www.slideshare.net/jimarlow/introductiontobpmn005
- BPMN v2.0 Specification
  - http://www.omg.org/spec/BPMN/2.0/PDF

SCRIPT: BPMN Training and Assessment

Please read through the entire script to familiarize yourself with the actions you'll be completing during this activity. Also, before you begin recording, open Notepad and the Windows Calculator, and resize/reposition them to be side-by-side on your screen. **Notepad on left, Calculator on right.

| Step	Action on Screen	Narration
1	*Built in narration to be cut out later during editing…*	Hello and welcome to this video tutorial on how to use the Business Process Model and Notation. **Business Process Model and Notation** (**BPMN**) is a graphical representation for specifying business processes in a business process model.
2	Notation	A BPMN process model is a graph consisting of four types of elements (among others): Flow Objects such as Events, Activities and Gateways Connecting Objects such as Sequence Flow and Message Flow Swimlanes such as Pool and Lanes (within a pool) Artefacts such as Data Object, Text Annotation and Group
3	Simple Order Management Process	This simple Order Management process is triggered by the reception of a purchase order from a customer. The purchase order has to be checked against the stock re the availability of the product(s) requested. Depending on stock availability the purchase order may be confirmed or rejected. If the purchase order is confirmed, the goods requested are shipped and an invoice is sent to the

		customer. In this diagram, plain gateways are used to denote points were the flow of control splits into multiple paths, and point where multiple paths converge into a single one. We can note that the first gateway in this diagram denotes a point where one among multiple paths needs to be chosen. Meanwhile, the second gateway denotes a point where two distinct paths need to be executed in parallel. Intuitively, it would make sense to distinguish between these two cases, that is, the case where one among many paths is chosen, and the case where multiple parallel paths are taken. In BPMN, this is shown by means of icons in the gateways. An "x" symbol in a gateway with multiple outgoing arcs indicates that one of multiple paths is taken. Such gateways are called exclusive gateways (remember that the "x" stands for exclusive). Meanwhile, a "+" symbol in a gateways with multiple outgoing arcs denotes that multiple parallel paths are executed. So the above diagram should be re-written as shown later…
4	Exclusive Gateways (XOR)	Exclusive Decision / Merge – Indicates locations within a business process where the sequence flow can take two or more alternative paths. – **Only one** of the paths can be taken. – Depicted by a diamond shape that *may* contain a marker that is shaped like an "X". – Exclusive merge Proceed when one branch has completed

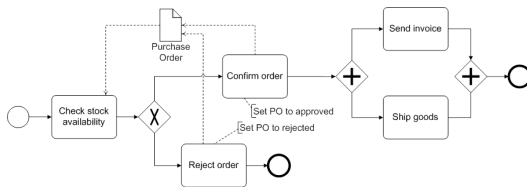| 5	*Parallel Gateway (AND)*	Parallel Fork / Join
		
		– Provide a mechanism to synchronize parallel flow and to create parallel flow.
		– Depicted by a diamond shape that *must* contain a marker that is shaped like a plus sign.
		– Parallel Fork
		   takes all branches
		– Parallel join
		   proceed when all incoming branches have completed

6	*Revised Order Management* *Process*	It is worth emphasizing here that activities located in two parallel paths do not need to be performed simultaneously. For example, "Send invoice" and "Ship goods" need not occur both at the same time, although due to a cosmic coincidence, they could happen at the same time. Instead, it might happen that first the invoice is sent and later the goods are shipped. Or things may happen in the reverse order.
7	Collaboration - Order Management Process with Pools	A **Collaboration** depicts the interactions between two or more business entities. A **Collaboration** usually contains two or more **Pools**, representing the *Participants* in the **Collaboration**. The **Message** exchange between the *Participants* is shown by a **Message Flow** that connects two **Pools** (or the objects within the **Pools**). The **Messages** associated with the **Message Flows** can also be shown. The Order Management process now includes the customer as a process participant… The Order Management process is started when a customer places a purchase order. The purchase order has to be checked against the stock re the availability of the product(s). Depending on stock availability the purchase

		order may be confirmed or rejected. If the purchase order is confirmed, the goods requested are shipped and an invoice is sent to the customer. The customer makes then makes the payment. Note that pools can be left partially unspecified.
8	Order Management Process with Lanes	The process now includes two departments within the supplier organization…The purchase order received by the Sales & Distribution department has to be checked against the stock. The order details are sent to the Warehouse department that returns an availability notification. If the purchase order is confirmed, the Warehouse department collects the shipping details from the customer and ships the goods. The Sales & Distribution department sends an invoice to the customer who then makes the payment.
9		*Data Objects* are a mechanism to show how data is required or produced by activities. The data object in this diagram is the Purchase Order. Associations are used to link artefacts such as data objects and data stores with flow objects (e.g. activities). The Purchase Order document serves as an input to the stock availability check. Based on the outcome of this check, the status of document is updated, either to "approved" or "rejected". We include here the relevant documents in the process

		model.
10		There are additional resources such as • Introduction to BPMN v2.0 by Dr. Jim Arlow • BPMN v2.0 Specification
		Thank you for taking the time to watch this video.

BPMN Assessment

The objective of this survey is to assess your level of skill with using the Business Process Management Notation.

TRINITY COLLEGE DUBLIN
INFORMATION SHEET FOR PARTICIPANTS

Many thanks for agreeing to participate in this assessment. The main objective of this assessment is to obtain information about your knowledge of the BPMN notation.

This assessment will take approximately 10 minutes.

Special Instructions:
- Participation in this study is voluntary; you may withdraw from the study at any time for any reason and omit questions you do not wish to answer without penalty.
- Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.
- Please do not name third parties in any open text field of the questionnaire. Any such replies will be anonymized.

Background of Research:

CRESUS (Collaborative Requirements Elicitation through enhancing Shared Understanding and Scenarios) addresses the need for a requirements elicitation approach that supports the collaboration among stakeholders during the communication of the organisation's requirements in a semantically consistent and understandable manner and then reflecting the potential impact of those requirements on the IT infrastructure of a business process.

The purpose of this study is to evaluate the effectiveness of the Communication and Modelling Tool, CRESUS-T at supporting collaborative requirements elicitation through enhancing the shared understanding of organisational and associated IT infrastructure requirements around communication and business process modelling controlling for level of

service, academic qualifications, and characteristics of subjects such as domain and modelling knowledge.

Procedures of this Research:

- You will be asked to fill in questionnaires before and after the experiment [10 minutes]
- An administrator will deliver a training session on the Business Process Modelling Notation (BPMN) followed by a questionnaire that examines your level of understanding of BPMN. [20 minutes].
- You will receive a description of the problem domain that represents some information provided during the early stages of requirements elicitation. Your task is to communicate asynchronously with other stakeholders in your team to develop a shared understanding of the IT infrastructure that will support the business process of the problem domain. This includes identifying any data stored in the system. You will be able to develop this understanding through the collaborative creation of a business process model, Stub IT implementation (prototype IT system) and data models. This part of the experiment will run over a one week period in a natural setting and in conjunction with your other work responsibilities. You will be requested to work on the experiment every day for at least 10 minutes. [At least 50 minutes]
- You will be asked to keep a journal during the experiment.
- You will be observed during the experiment.
- Data will be logged during the experiment.
- You will be interviewed on questions that relate to the experiment and to clarify any data collected during the experiment. The interview will be recorded. [15 minutes]
- You will be debriefed on the experiment. [15 minutes]
- The total duration of the study should be approximately 1.75 hours over a 2 week period.

Participant Selection Procedure:

• Participants will be selected from the School of Computing and IT Department.

Health and Safety:

• If you have a history of Photosensitive Epilepsy or any other condition that may cause you to experience difficulty with this experiment, please inform the supervisor of the study.

Conflict of Interest:

Conflict of Interest:

• Please be advised that this research is being conducted by an employee of the college which may involve taking advantage of an existing relationship in order to progress this research.

Legal Notes:

• All information collected for this study will be recorded anonymously and held confidentially to protect your privacy. All information gathered will be used solely for this study.

• Any data used in publications will be aggregated and no personally identifiable information we revealed.

• Participation in this study is voluntary; you may withdraw from the study at any time for any reason and omit questions you do not wish to answer without penalty.

• In the extremely unlikely event that illicit activity is reported to me during the study I will be obliged to report it to appropriate authorities.

If you would like further details about the study feel free to contact Paul Stynes using the details below.

Name: Paul Stynes Email:pstynes@ncirl.ie Tel: 01 449 8613 Address: National College of Ireland, Mayor Street, Dublin 1.

There are 14 questions in this survey

Participant Identification

1   Please enter the code assigned to you. *

Please write your answer here:

Level of Confidence

2   Rate your confidence in using Business Process Model and Notation (BPMN). *

Please choose **only one** of the following:

- ○not confident
- ○somewhat confident
- ○average confidence
- ○confident
- ○very confident

Modelling understanding

The following questions test your understanding of the Business Process Modelling Notation

3 Please   select   a   word   that   represents   the   symbol

◯ .

Please choose **only one** of the following:

- ○Event
- ○Activity
- ○Gateway
- ○Sequence Flow
- ○Message Flow
- ○Association
- ○Pool
- ○Lane (within a pool)

4 Please   select   a   word   that   represents   the   symbol

▢

Please choose **only one** of the following:

- ○Event
- ○Activity
- ○Gateway
- ○Sequence Flow
- ○Message Flow
- ○Association
- ○Pool
- ○Lane (within a pool)

5 Please   select   a   word   that   represents   the   symbol

Please choose **only one** of the following:

- ○Event
- ○Activity
- ○Gateway
- ○Sequence Flow
- ○Message Flow
- ○Association
- ○Pool
- ○Lane (within a pool)

6 Please   select   a   word   that   represents   the   symbol

Please choose **only one** of the following:

- ○Event
- ○Activity
- ○Gateway
- ○Sequence Flow
- ○Message Flow
- ○Association
- ○Pool
- ○Lane (within a pool)

7 Please   select   a   word   that   represents   the   symbol

Please choose **only one** of the following:

- ○Event
- ○Activity
- ○Gateway
- ○Sequence Flow
- ○Message Flow
- ○Association

- ○ Pool
- ○ Lane (within a pool)

8 Please   select   a   word   that   represents   the   symbol

Please choose **only one** of the following:

- ○ Event
- ○ Activity
- ○ Gateway
- ○ Sequence Flow
- ○ Message Flow
- ○ Association
- ○ Pool
- ○ Lane (within a pool)

9 Please   select   a   word   that   represents   the   symbol

Please choose **only one** of the following:

- ○ Event
- ○ Activity
- ○ Gateway
- ○ Sequence Flow
- ○ Message Flow
- ○ Association
- ○ Pool
- ○ Lane (within a pool)

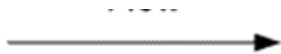**10 Please   select   a   word   that   represents   the   symbol**

Please choose **only one** of the following:

- ○ Event
- ○ Activity
- ○ Gateway

- ○ Sequence Flow
- ○ Message Flow
- ○ Association
- ○ Pool
- ○ Lane (within a pool)

11 Please select a word that represents the symbol
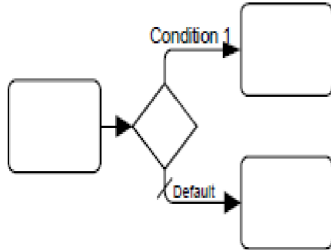
Please choose **only one** of the following:

- ○ Parallel Gateway Fork
- ○ Parallel Gateway Join
- ○ Decision Branching Point - Exclusive Database Gateway
- ○ Merging Exclusive Database Gateway

12 Please select a word that represents the symbol
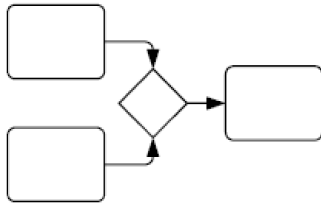
Please choose **only one** of the following:

- ○ Parallel Gateway Fork
- ○ Parallel Gateway Join
- ○ Decision Branching Point - Exclusive Database Gateway
- ○ Merging Exclusive Database Gateway

13 Please select a word that represents the symbol
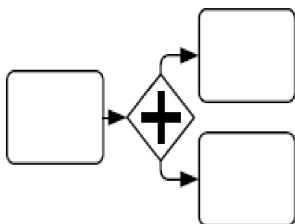
Please choose **only one** of the following:

- • ◯Parallel Gateway Fork
- • ◯Parallel Gateway Join
- • ◯Decision Branching Point - Exclusive Database Gateway
- • ◯Merging Exclusive Database Gateway

14 Please   select   a   word   that   represents   the   symbol
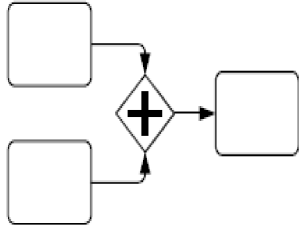
Please choose **only one** of the following:

- • ◯Parallel Gateway Fork
- • ◯Parallel Gateway Join
- • ◯Decision Branching Point - Exclusive Database Gateway
- • ◯Merging Exclusive Database Gateway

SCRIPT: CRESUS-T Training

Please read through the entire script to familiarize yourself with the actions you'll be completing during this activity. Also, before you begin recording, start CRESUS.

|	Action on Screen	Narration
1	*CRESUS-T*	Hello and welcome to this video tutorial on how to use CRESUS-T for collaboratively eliciting potential organizational and IT infrastructural requirements through enhancing shared understanding around - communication -business process modelling of a scenario
2	CRESUS Approach	The CRESUS approach (see Figure 1) incorporates several techniques for requirements elicitation namely: • Interview • brainstorming which is a group elicitation technique: • Controlled Natural Language • Use of prototype and stub implementation which is an artefact which stakeholders can respond to. • Modelling techniques • Scenarios The techniques are combined with process guidance for the stakeholders. The approach consists of four phases: 1. Phase 1 Problem Understanding. This phase consists of an interview with the business executive to get an understanding of how the process works and the issues with the current process. 2. Phase 2 Domain modelling. This is conducted as a brainstorming session where the stakeholders define the concepts and key relationships of the problem domain. These concepts and relationships form part of

the concept model and actions model. For example Clerk issues Registration represents concepts Clerk and Registration. issues represents an action. The conceptual model is then approved by all stakeholders. The business executive and users are the key stakeholders during this phase.

3. The requirements elicitation forms phase 3 and consists of two sub phases namely the requirements gathering phase and the generation and testing phase.

    a. Requirements Gathering phase incorporates controlled natural language as an elicitation technique. The primary stakeholders that are active in this phase are the business executive and users of the system. They define the instance data for example JohnSmith issues Registration102455. There is an incremental approval process in place. Each statement that is gathered is approved by the other stakeholders. The approved statement forms a triple of CNL requirements that form part of the instance model. This phase concludes when all stakeholders indicate that they have completed the phase.

    b. The generation and testing phase incorporates stub implementation as a requirements elicitation technique. The stub implementation is automatically generated from the concept model, instance model and actions. The stub implementations have limited functionality and interactivity. Concepts such as Clerk is mapped to web services. Actions such as issues are mapped to operations in the web service. The last concept of the triple such as Registration is mapped to parameter data of the operation. A series of discrete web services that form the IT Stub Architecture are created and deployed on the server. The instance model is transformed into a series of XML files that are stored in an online database. The stakeholders especially the IT architect will be able to critique the web services, interfaces and data models. The Stakeholders are given a testers view of the deployed web services where they can add additional instance data to the instance model through the web service api. This phase

		concludes when all stakeholders indicate that they have completed the phase. 4 The 4th Phase Business Process Modelling incorporates modelling a scenario through controlled natural language. All stakeholoders are active in this phase. This phase combines the buisness executive and users view of the process with the IT architectures view of the stub implementations. The web services are mapped to pools in the business process model. A scenario is created that incorporates intelligence thorugh tasks and sequencing of the process activities. The artefacts produced in this phase are the Business Process Model and IT Stub Architecture. The Business Process Model can be mapped to an executable procedd language such as BPEL and deployed on a server.
3	**Let's look at an example Vehicle Registration System**	The Texas vehicle registration system allows owners to register new vehicles such as a car, truck, motorcycle and van. The vehicle contains details such as their name, address, phone number, license number, make, model, fuel type, colour and the year the vehicle was manufactured. A clerk at the vehicle registration department can verify the owner of a vehicle. The truck can be either commercial or non-commercial. For a fee the owner is issued a new registration. For the registration of older vehicles, the system calculates a renewal fee. For commercial trucks, the owner is required to confirm an emission check which gives an emission value. The owner is then issued a renewal notice. The owner can then make an online payment.
4	Concepts from Problem Domain	By analysing the problem domain we can identify the following concepts • Clerk • Fee • Owner • Payment • Registration • Vehicle

		− Car
		− Motorbike
		− Van
		− Truck
		• Commercial Truck
		• Non-commercial Truck
5	*Actions*	From the problem domain we can identify the following actions
		• calculatesRenewalFee
		• confirmsEmissionCheck
		• isIssuedTo
		• issuesNewRegistration
		• issuesRenewalNotice
		• makesOnlinePayment
		• Owns
		• Provides
		• verifiesOwnership
		• hasAddress
		• hasAmount
		• hasColour
		• hasCreditCardNumber
		• hasCreditCartType
		• hasFuelType
		• hasLicenceNumber

		• hasManufacturerMake
		• hasManufacturerModel
		• hasYearOfManufacture
6	*Instance Data*	To understand the instance data we take a look at the controlled natural language requirements that form a triple in terms of the concept action and concept.
		For example:
		Owner hasLicenceNumber integer
		JaneDoe hasLicenceNumber 234222222.
		Motorbike hasFuelType String
		Motorbike1 hasFuelType Petrol.
		Motobike hasManufacturerModel String
		Motorbike1 hasManufacturerModel GoldWing.
7	Approval Process	Each CNL requirement will go through an approval process by the stakeholders. The stakeholders are represented by three roles in the system namely
		– Business Executive
		– IT Architect
		– Users
		If the Business Executive creates a requirement then according to the approval process the requirement will go to the User for approval. If approved then it will go the IT architect. Likewise
		– Users -> Business Executive -> IT Architect

		• Example
		
		   – UserA submits the following data "Van1 hasColour Red."
		
		   – Data goes to Business Executive for approval
		
		   – If approved then it goes to the IT Architect for approval
		
		   – If approved the data then forms part of the Data Model

8	Metrics - Communication	• Frequency
		
		   – Business Executive enters "Motorbike1 hasYearOfManuafacuture 2001."
		
		        • Frequency of communication = 1
		
		   – User approves data
		
		        • Frequency of communication = 2
		
		   – IT architect approves data
		
		        • Frequency of communication = 3
		
		• Quality of Communication
		
		   – Data becomes part of data model
		
		        • Quality of message (Identifying a solution)= 1

9	Metrics – Shared Understanding	• *Model Recall* • *Model Understanding* • *Problem solving question around the model*
10	Metrics - Opinion	• Communication • Business Process Modelling • Shared Understanding

		• Approach to Collaborative Requirements Elicitation
11	CRESUS-T – Data Entry User logs in	• Login • Instance Data • Table • Create Instance Data • Exit • Complete online journal
12	Business Executive logs in	• Login • Rate your understanding of the created data • Approve Changes • Instance Data • Create Instance Data • Exit • Complete online journal
13	IT Architect logs in	• Login • Rate your understanding of the created data • Approve Changes • Instance Data • Create Instance Data • Exit • Complete online journal

14	User logs in	• Login
		• Rate your understanding of the created data
		• Approve Changes
		• Instance Data
		• Data Entry Complete
		• Exit
		• Complete online journal
15	Business Executive logs in	• Login
		• Rate your understanding of the created data
		• Approve Changes
		• Instance Data
		• Data Entry Complete
		• Exit
		• Complete online journal
16	IT architect logs in	• Login
		• Rate your understanding of the created data
		• Approve Changes
		• Instance Data
		• Data Entry Complete
		• Exit
		• Complete online journal
17	CRESUS-T – Create IT Architecture IT architect logs in	• Login
		• View IT Architecture
		• Test Web Service

		• View the Interface
		• View the Data Model
		• Test Web Service
		• Get All Web Service Names
		• Set Web Service Name
		• Add some data through the API
		• View the Data Model
		• Select IT Architecture Complete
		• Exit
		• Complete online journal
18	Business Executive logs in	• Login
		• View IT Architecture
		• Test Web Service
		• View the Interface
		• View the Data Model
		• Test Web Service
		• Get All Web Service Names
		• Set Web Service Name
		• Add some data through the API
		• View the Data Model
		• Select IT Architecture Complete
		• Exit
		• Complete online journal
19	User logs in	• Login

		• View IT Architecture
		• Test Web Service
		• View the Interface
		• View the Data Model
		• Test Web Service
		• Get All Web Service Names
		• Set Web Service Name
		• Add some data through the API
		• View the Data Model
		• Select IT Architecture Complete
		• Exit
		• Complete online journal
20	CRESUS-T – Business Process Model Business Executive logs in	• Login • Business Process • Business Process Diagrams • Pool • Process • BPMN Elements • Exit • Complete online journal
21	IT Architect logs in Focus is on the web services mapped to pools Actions/API operations mapped	• Login • Business Process • Business Process Diagrams • Pool

	to tasks	• Process
		• BPMN Elements
		• Exit
		• Complete online journal
22	User logs in	• Login
		• Business Process
		• Business Process Diagrams
		• Pool
		• Process
		• BPMN Elements
		• Exit
		• Complete online journal
23	Business Executive logs in	• Login
		• Business Process
		• Business Process Diagrams
		• Pool
		• Process
		• BPMN Elements
		• Business Process Complete
		• Exit
		• Complete online journal
24	IT Architect logs in	• Login
		• Business Process
		• Business Process Diagrams
		• Pool

		• Process
		• BPMN Elements
		• Business Process Complete
		• Exit
		• Complete online journal
25	User logs in	• Login
		• Business Process
		• Business Process Diagrams
		• Pool
		• Process
		• BPMN Elements
		• Business Process Complete
		• Exit
		• Complete online journal
26		This concludes our session on BPMN.
		Thank you for taking the time to watch this video.

CRESUS Post test

Many thanks for agreeing to participate in this survey. The main objective of this survey is to obtain information about your - Model Recall - Model Understanding - Problem Solving

This survey will take approximately 10 minutes.

Special Instructions:

- Participation in this study is voluntary; you may withdraw from the study at any time for any reason and omit questions you do not wish to answer without penalty.

- Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

- Please do not name third parties in any open text field of the questionnaire. Any such replies will be anonymised.

Background of Research:

CRESUS (Collaborative Requirements Elicitation through enhancing Shared Understanding and Scenarios) addresses the need for a requirements elicitation approach that supports the collaboration among stakeholders during the communication of the organisation's requirements in a semantically consistent and understandable manner and then reflecting the potential impact of those requirements on the IT infrastructure of a business process. The purpose of this study is to evaluate the effectiveness of the Communication and Modelling Tool, CRESUS-T at supporting collaborative requirements elicitation through enhancing the shared understanding of organisational and associated IT infrastructure requirements around communication and business process modelling controlling for level of service, academic qualifications, and characteristics of subjects such as domain and modelling knowledge.

Procedures of this Research:

• You will be asked to fill in questionnaires before and after the experiment [10 minutes] • An administrator will deliver a training session on the Business Process Modelling Notation (BPMN) followed by a questionnaire that examines your level of understanding of BPMN. [20 minutes]. • You will receive a description of the problem domain that represents some information provided during the early stages of requirements elicitation. Your task is to communicate asynchronously with other stakeholders in your team to develop a shared understanding of the IT systems that will support the business process of the problem domain. This includes identifying any data stored in the system. You will be able to develop this understanding through the collaborative creation of a business process diagram, prototype IT system and data models. This part of the experiment will run over a one week period in a natural setting and in conjunction with your other work responsibilities. You will be requested to work on the experiment every day for at least 10 minutes. [At least 50 minutes] • You will be asked to keep a journal during the experiment. • You will be observed during the experiment. • Data will be logged during the experiment. • You will be interviewed on questions that relate to the experiment and to clarify any data collected during the experiment. The interview will be recorded. [15 minutes] • You will be debriefed on the experiment. [15 minutes] • The total duration of the study should be approximately 1.75 hours over a 2 week period.

Procedure:

• Participants will be selected from the School of Computing and IT Department. Health and Safety:

• If you have a history of Photosensitive Epilepsy or any other condition that may cause you to experience difficulty with this experiment, please inform the supervisor of the study.

Conflict of Interest:

• Please be advised that this research is being conducted by an employee of the college which may involve taking advantage of an existing relationship in order to progress this research.

Legal Notes:

• All information collected for this study will be recorded anonymously and held confidentially to protect your privacy. All information gathered will be used solely for this study.

• Any data used in publications will be aggregated and no personally identifiable information we revealed.

• Participation in this study is voluntary; you may withdraw from the study at any time for any reason and omit questions you do not wish to answer without penalty.

• In the extremely unlikely event that illicit activity is reported to me during the study I will be obliged to report it to appropriate authorities.

If you would like further details about the study feel free to contact Paul Stynes using the details below.

Name: Paul Stynes Email: pstynes@ncirl.ie Tel: 01 449 8613 Address: National College of Ireland, Mayor Street, Dublin 1.

There are 3 questions in this survey

Participant Identification

1 Please enter the code assigned to you. *

Please write your answer here:

Model Recall

2 In order to assess your recollection of the experiment, please provide a business process model of the deployment of the IT image. *

Please write your answer here:

Problem Solving

3   NCI would like to improve the process for deploying the IT image to the classroom by including a scenario that allows the installation of customised software on all the pc's in a specific classroom. Could you please create a business process model that handles this scenario? *

Please write your answer here:

Interview with Stakeholders

Communication

1. Describe your experience of asynchronous communication around completing the data model, IT infrastructure and business process model
2. How would you improve communication during the study
3. Describe your experience with using Natural Language part of the CRESUS-T tool

Business Process Modelling

1. Describe your experience with collaborative business process modelling
2. Describe your experience with using the Natural Language part of the CRESUS-T tool for modelling the Business Process

Shared Understanding

1. Describe your experience with the approach of reaching a shared understanding of the data model, IT infrastructure and business process model

Collaborative Requirements Elicitation

1. Describe your experience with collaboration during the creation of the data model, IT infrastructure and business process model
2. In your opinion does the IT infrastructure have the potential to represent a real world solution

Appendix 4.　Evolution of Themes

The themes that emerged from the experimental study three, arising out of the interviews and journal are described as follows:-

Table A4-1 Evolution of themes arising out of interviews and journal

Question	Category	Theme		
Describe your experience of asynchronous communication around completing the data model, IT infrastructure and business process model	Negative experience with asynchronous communication	Lack of Understanding		
		Limitations of communication tool		
		Little or no communication		
		Requirements not approved		
		Peoples availability		
		Tasks drift		
		Technical issues		
	Positive experience with asynchronous communication	People are connected		
		Productive activities		
		Intuitive		
		Requirements to approve		
Describe your experience with collaboration during the creation of the data model, IT infrastructure and business process model	Negative experience with collaboration	Asynchronous communication		
		Bilateral communication		
		Lack of understanding		
		Limited or no participation		
		Working independently		
	Positive experience with collaboration	Better Understanding of Others	Collaboration	
		Tool facilitates collaboration		
		Decision Making		
		Clear and understandable approach	Clear and understandable approach	
		Useful feedback	Useful feedback	
Describe your experience with collaborative	Negative Experience	Did not model		
		Modelling Independently		

business process modelling		Requires a specialist role or practice		
	Positive Experience	Visibility of collaboration and problem solving		
		Easy to use the interface		
		Liked the model		
Describe your experience with the approach of reaching a shared understanding	Negative experience with approach of reaching a shared understanding	Independent understanding		
		Lack of coordination		
		Lack of understanding		
		Work commitments		
		Approval process		
		illness		
		Use of asychronous communication		
	Positive experience with approach of reaching a shared understanding	Same tool same modelling language		
		Use of concrete data		
		Collaboration and understanding		
Describe your experience with using Natural Language part of the CRESUS-T tool	Negative experience with using natural language part of the CRESUS-T tool	Instance data not approved		
		Lack of flexibility		
		Did not use		
	Positive experience with using natural language part of the CRESUS-T tool	Intuitive		
		Use of natural language		
Describe your experience with using the Natural Language part of the CRESUS-T tool for modelling the	Negative experience with using natural language for modelling the business process	difficulty with naming selecting understanding language	difficulty with naming selecting understanding language	
		No participation	Requires Practice	
		Technical issue		
		Did not use	Did not use	

Business Process	Positive experience with using natural language for modelling the business process	Build up a system	Build up a system	
		Useful	Useful	
How would you improve communication	Additional communication channels	Additional communication channels	Additional communication channels	
	Event based message system			
	Face to face meetings			
	Forum			
	Mail Server			
	Messenger service			
	Online chat			
	Reminder emails			
	Sharing software			
	Commitment to engage in the process	Commitment to engage in the process	Commitment to engage in the process	
	Detailed Instruction	Detailed Instruction	Detailed Instruction	
	Formal checkpoints	Formal checkpoints	Structured approach	
	Phases mapped to time bound deadlines	Phases mapped to time bound deadlines		
	Time bound meetings	Set time		
	Set time			
	Strong leadership	Strong leadership	Stronger leadership	
In your opinion does the IT infrastructure have the potential to represent a real	Negative opinion	Confusion		
		Did not communicate ideas		
		Did not understand		

world solution				
		Limited or no participation	No	
		No		
	No opinion			
	Positive opinion	Geographically distributed collaboration		
		Biuilding blocks to support business processes		
		Yes		
		Sandbox		
		Realistic Scenario		
Journal	Challenges	Confusion	Confusion	Confusion
		How to create data	Creating artefacts	Creating artefacts
		How to model a business process		
		Problem domain understanding	Problem domain understanding	Problem domain understanding
		Problem was not relevant		
		Loosing interest		
		Missing data	suitable software environment	suitable software environment
		No access to software		
		No group email		
		No visualisation of data model		
		Software environment issues		
		Unable to create relationships and concepts		
		Use natural language editor		
		Leadership	Leadership	Participation
		Participation	Participation	
		Time constraints		
		Approval process for instance data	Understanding of approach	Understanding of approach
		Timing between phases		
	Highlights	Better understanding of problem	Collaboration	Collaboration
		Group Communication		
		Good collaboration		

			Creating requirements	Creating requirements	Creating requirements
			Easy to understand natural language	Easy to understand natural language	Easy to understand natural language
			Easy to use interface	Software environment	Software environment
			Easy to use software system		
			Easy to view instance data		
			Exploring new software environments		
			Knowledge of using software environment		
			Familiarity with modelling tools		
			Interaction with IT Architecture		
			Use of approval system		
			Understood IT Architecture	Understood IT Architecture	Understood IT Architecture
			Making progress	Making progress	Making progress
			Informative Tutorials	Case study and tutorials	Case study and tutorials
			Well documented problem domain		